

# SIMATIC S7-1200

---

KROK ZA KROKEM



Autor: Ing. Josef Havel

## Obsah

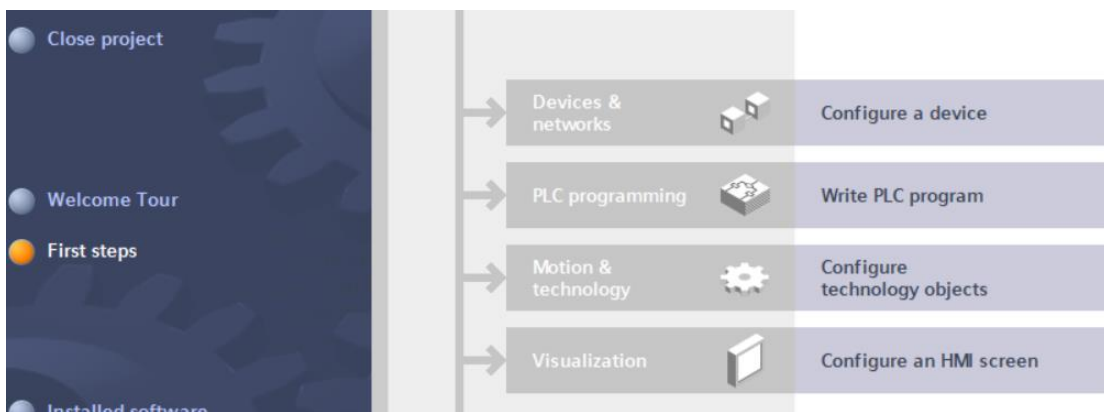
1.	VYTVOŘENÍ PROJEKTU V TIA PORTÁLU .....	2
2.	Přidání dotykového panelu HMI.....	4
3.	Simulace.....	6
4.	Nastavení PLC tags - nastavení adres proměnných.....	10
5.	Data block .....	10
6.	Otevření programovací plochy.....	14
7.	Programování v LAD.....	14
8.	Logické bitové bloky.....	20
9.	Časové bloky .....	32
10.	Čítače .....	44
11.	Komparátory .....	48
12.	Matematické funkce .....	53
13.	Programování v FBD .....	84
14.	Analogové funkce .....	91
15.	Propojení s panelem HMI.....	100
	.....	104
	.....	105
16.	PŘÍKLADY .....	109
a)	Postupné spínání .....	109
b)	Postupné spínání a vypínání – první se zapíná i vypíná Q1.....	111
c)	Postupné spínání a vypínání – první se zapíná Q1 a první se vypíná Q2 .....	111
e)	Pulzní generátor .....	115
f)	Střídavé spínání a vypínání dvou výstupů.....	117
g)	Rozběh motoru hvězda trojúhelník.....	118
h)	Krokový posun.....	119
ch)	Cyklické spínání výstupů .....	120
i)	Cyklické spínání výstupů s konečným počtem cyklů .....	121
j)	Větrání nebo chlazení .....	122
l)	Měření spouštění a jistění asynchronního motoru.....	127
m)	Semafor.....	136

# 1. VYTVOŘENÍ PROJEKTU V TIA PORTÁLU

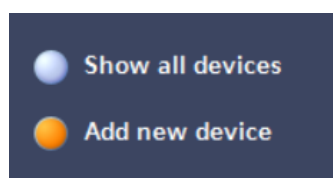
1. Klik LT na „Create new projekt“



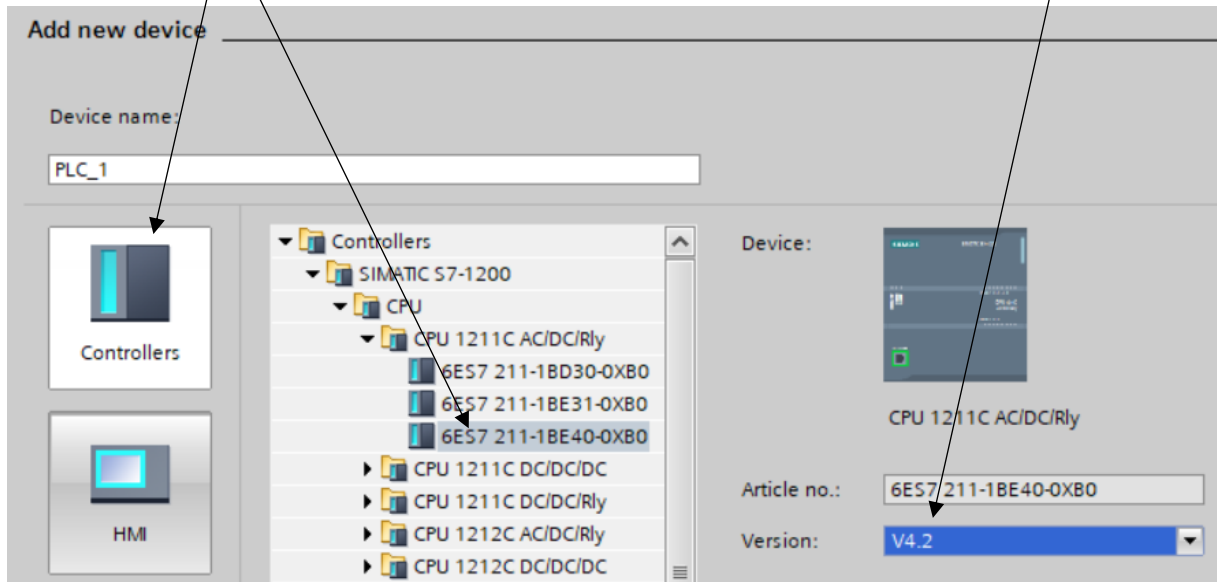
2. Klik LT na „Configure a device“



3. Označíme „Add new device“ – přidat nový přístroj

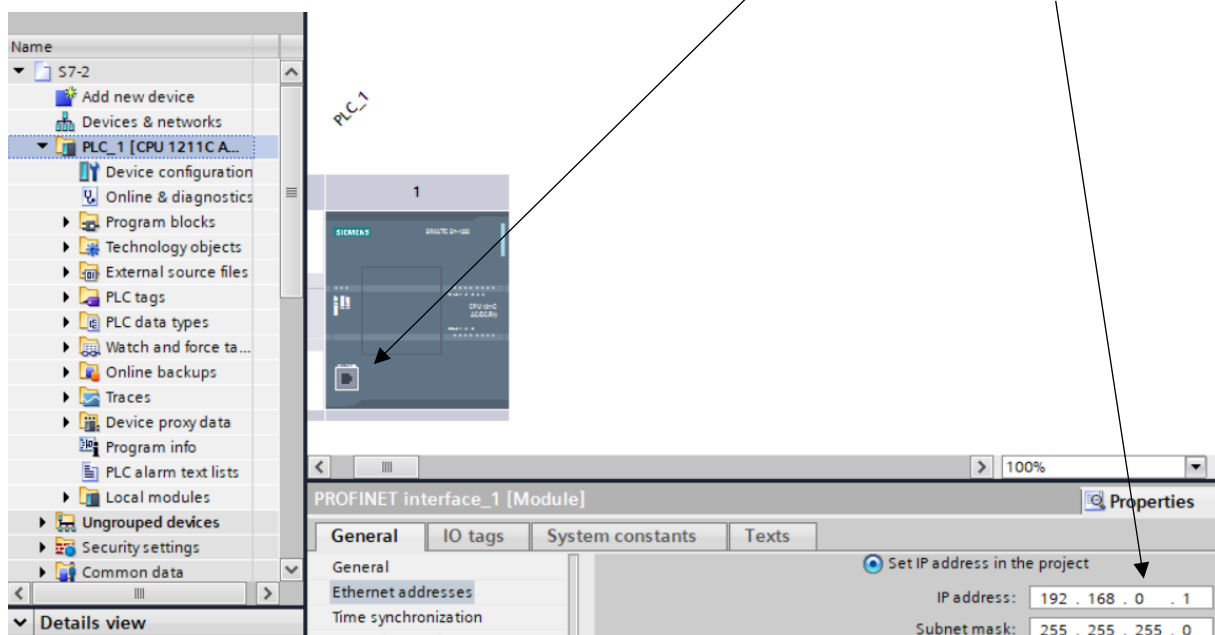


4. Kliknutí na „Controllers“ se otevře nabídka přístrojů.
5. Vybereme přístroj. Pokud programujeme ve verzi 15.1 potom musí být verze FW V4.2, jinak nebude fungovat simulace.



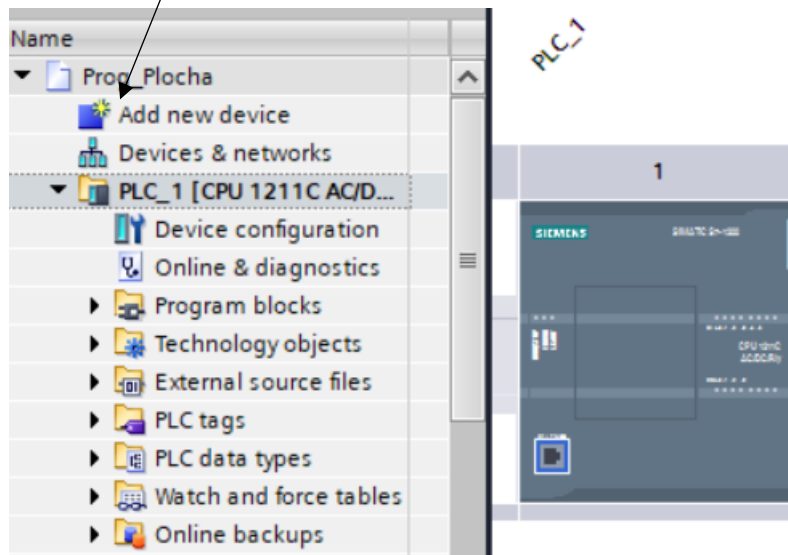
6. Klik LT na Add

7. Otevře se ovládací panel s blokem PLC. Kliknutím na datový výstup se zobrazí adresa přístroje

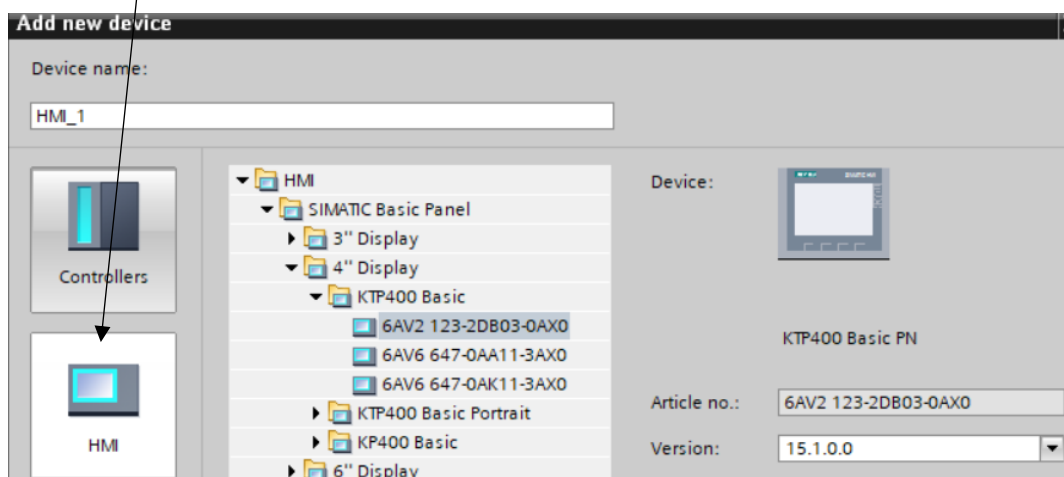


## 2. Přidání dotykového panelu HMI

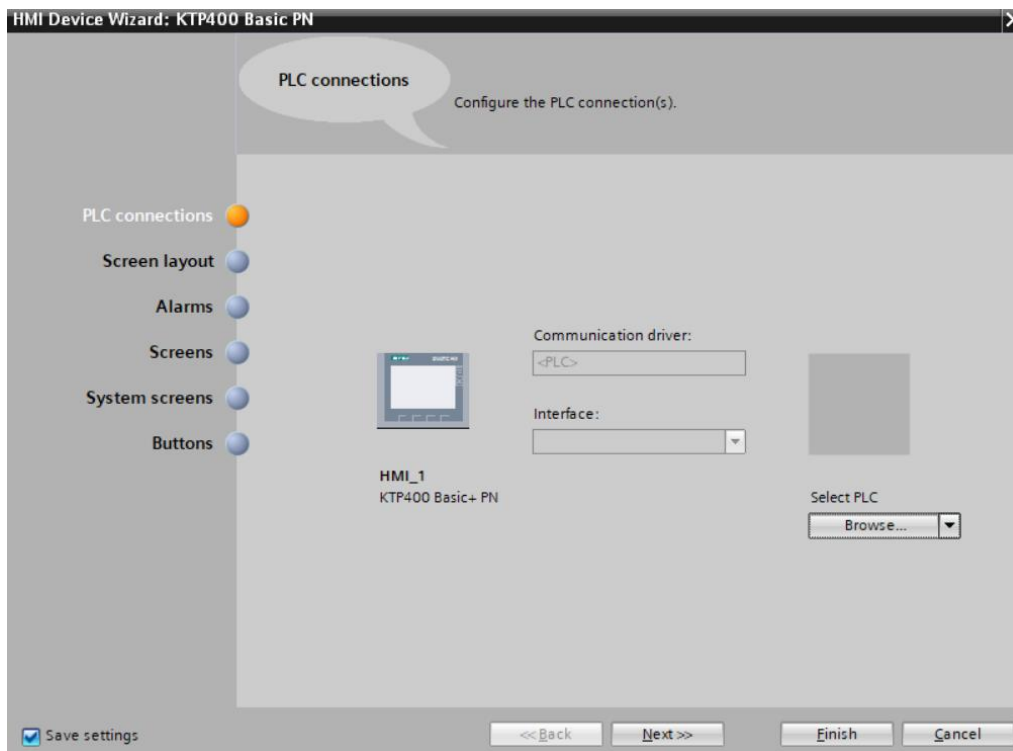
1. Klik LT na „Add new device“



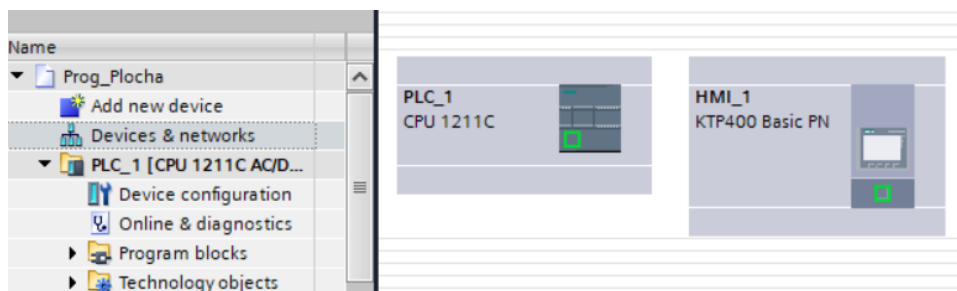
2. Klik na HMI



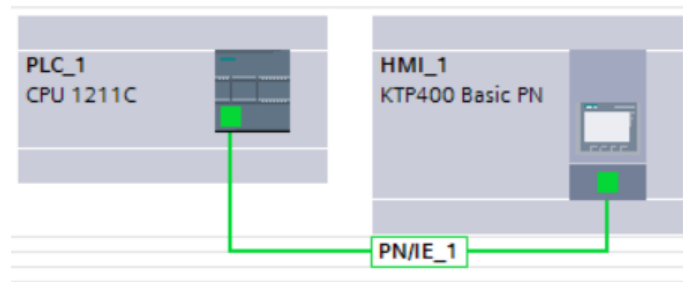
3. Vybereme panel, který budeme programovat
4. Klik LT na Add
5. Klik LT na Browse se připojí panel HMI k přístroji S1200
6. Klik LT na „Finish“



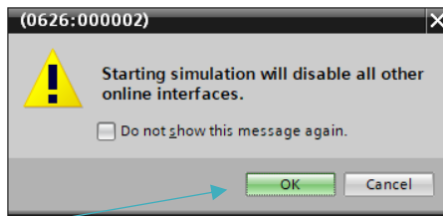
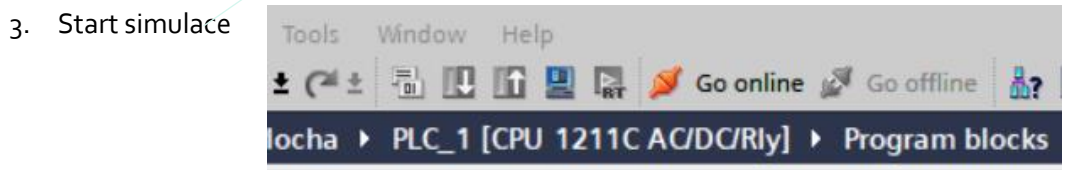
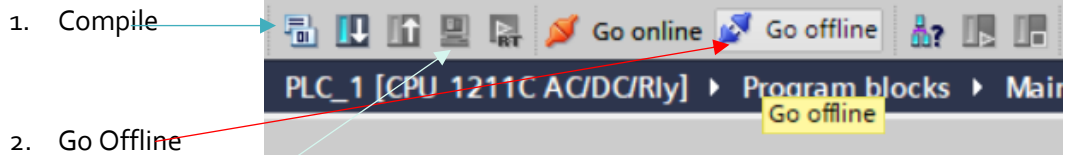
Pokud neprovedeme propojení v ikoně Browse provedeme Klik LT na „Devices & device“



Klik LT na výstup z PLC\_1 a táhnout myší na vstup HMI\_1.

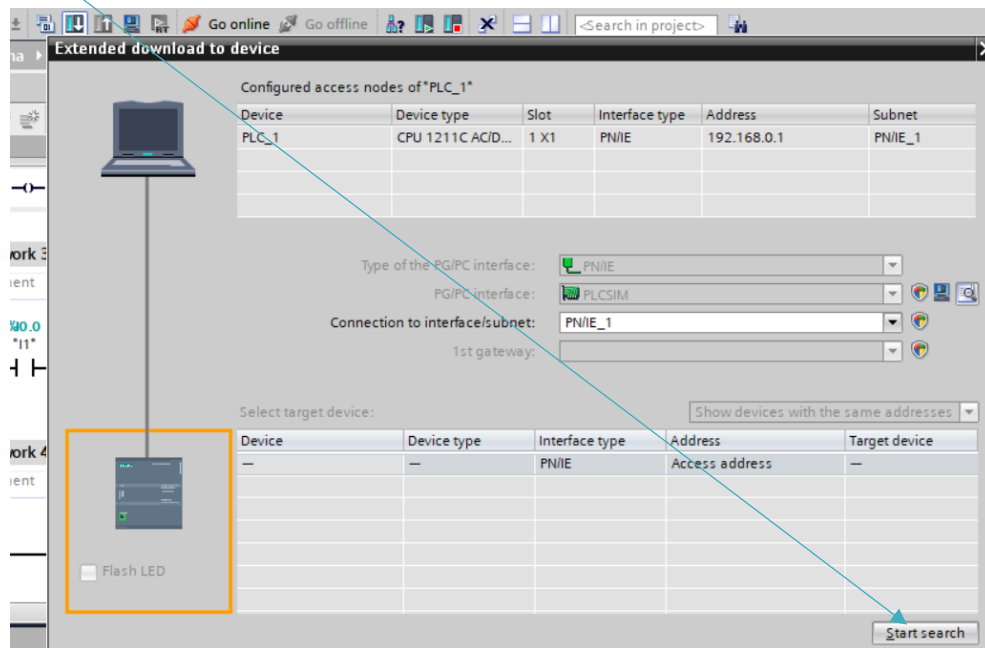


### 3. Simulace

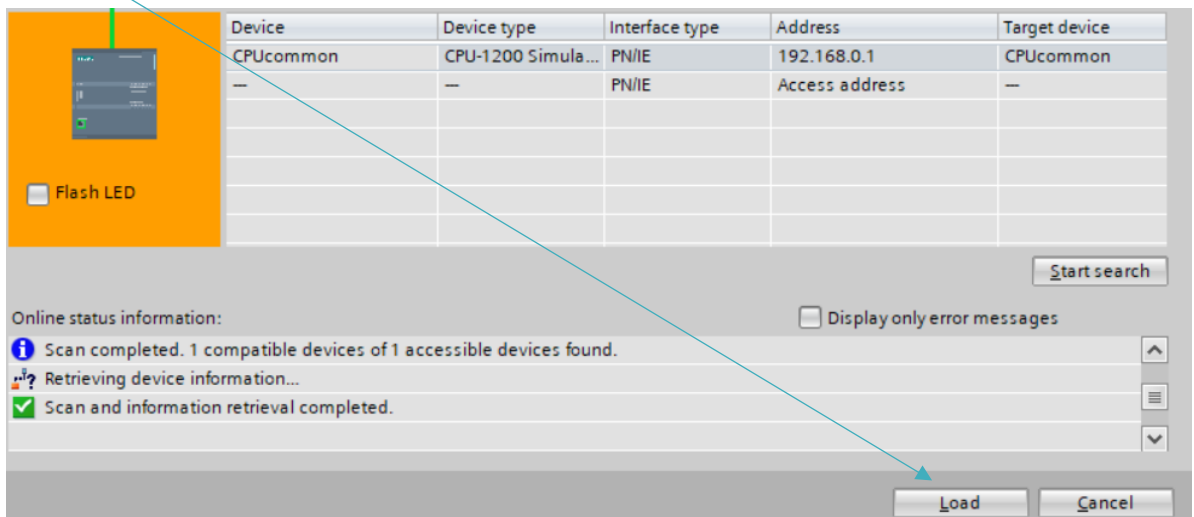


4. OK →

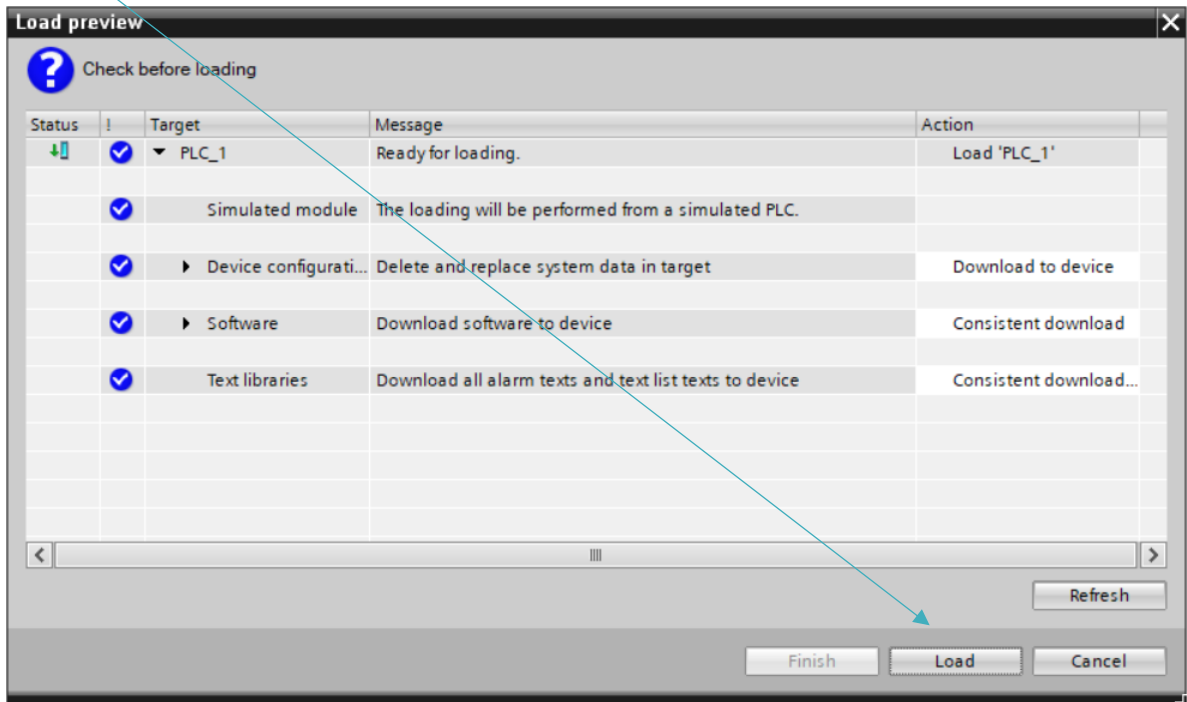
5. Start search – vyhledání programu



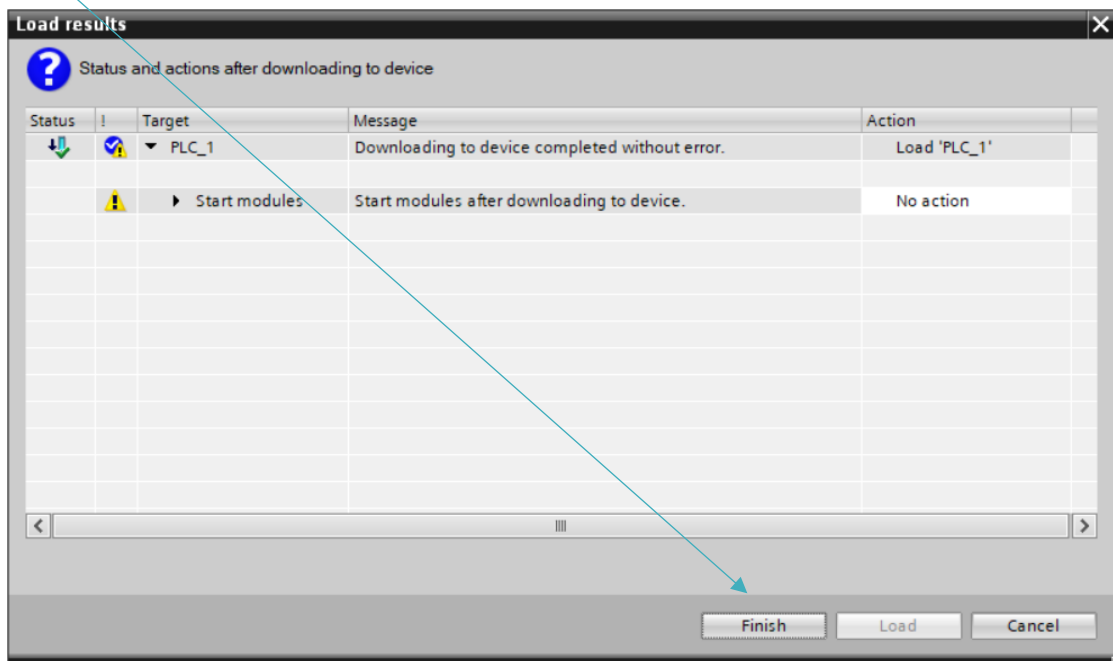
6. Load (nahrát)



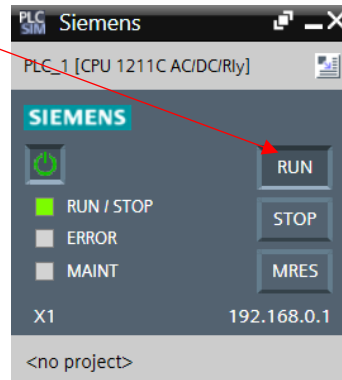
## 7. Load



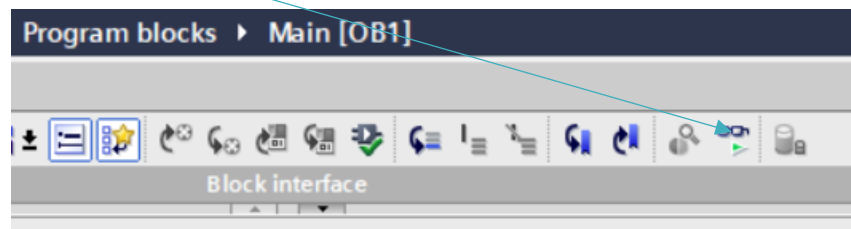
## 8. Finish (dokončit)



9. Klik LT na RUN



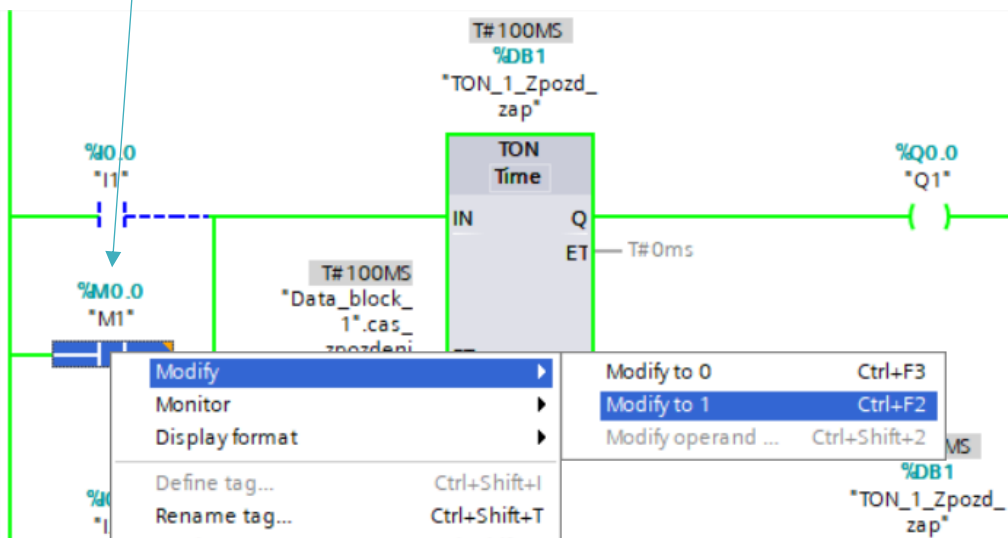
10. Klik LT  na



11. Klik **PT** na blok, který budeme spínat

12. Klik LT na Modify (změnit)

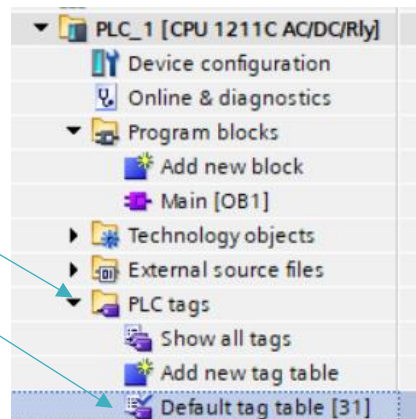
13. Klik LT na Modify 1 (sepnutí), nebo Modify 2 (vypnutí)



Vstupy I nelze simulovat. Simulaci lze provádět se vstupy M

## 4. Nastavení PLC tags - nastavení adres proměnných

1. Klik LT na šipku „PLC tags“
2. klik LT na „Default tag table“
3. Zapiší se vstupy, výstupy a jejich adresy



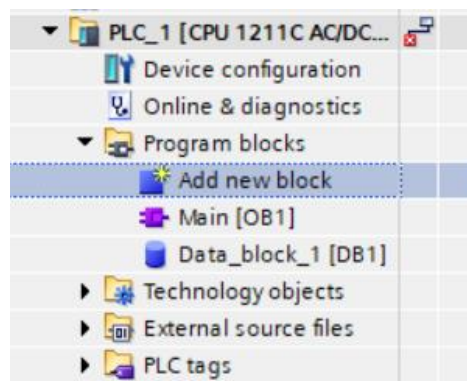
The screenshot shows the 'Default tag table' configuration window. It contains a table with columns for 'Name', 'Data type', and 'Address'. The table is populated with three entries: 'I1\_zap' (Bool, %I0.0), 'I2\_vyp' (Bool, %I0.1), and 'Q1\_vystup' (Bool, %I0.2). A fourth row contains '<Add new>'. The left sidebar shows the same tree view as the previous image, with 'Default tag table [31]' selected.

	Name	Data type	Address
1	I1_zap	Bool	%I0.0
2	I2_vyp	Bool	%I0.1
3	Q1_vystup	Bool	%I0.2
4	<Add new>		

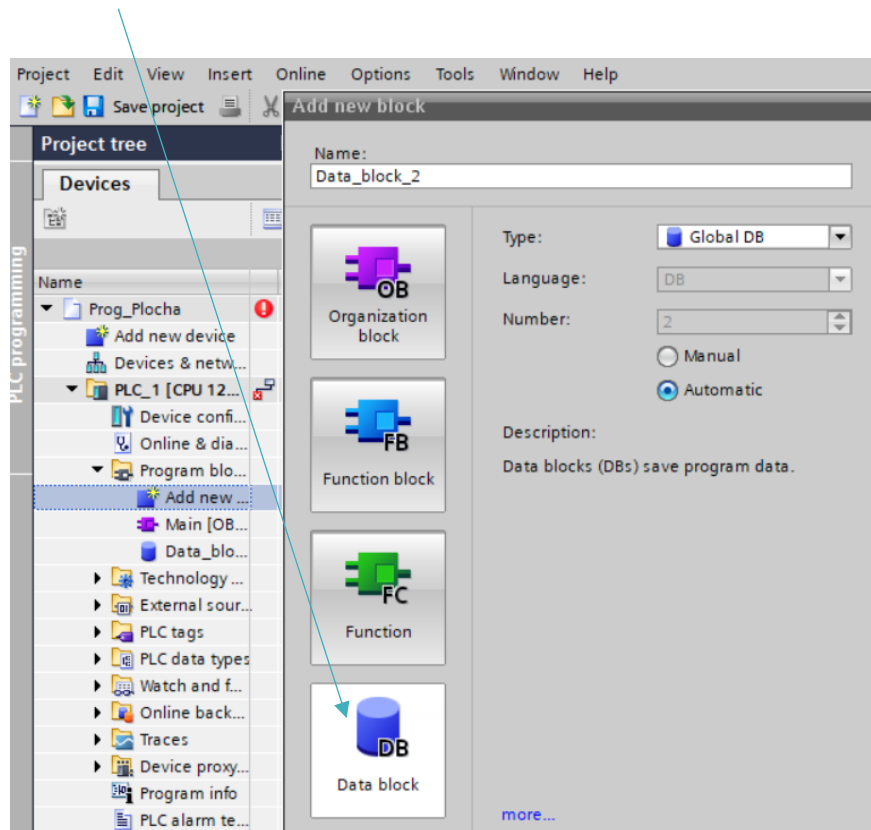
## 5. Data block

Nastavení proměnných lze provést nejen v PLC tag, ale i v „Data\_block“.

1. Klik LT na „Add new block“



2. Klik LT na „Data block“



3. Klik LT na OK – otevře se tabulka, do ní zapíšeme název proměnné a datový typ

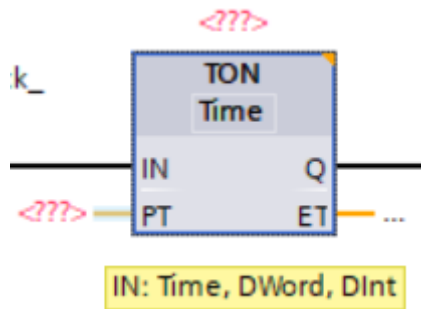
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...
1	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<Add new>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Chceme-li adresu vložit nad některý již zapsaný řádek (M1) klikneme LT na hvězdičku „Insert row“ (vložit řádek).

	Name	Data type	Start value
1	Static		
2	<Add new>		
3	M1_Start	Bool	false
4	M2_Stop	Bool	false
5	<Add new>		
6	Q1_Vystup	Bool	false

Vložení řádku pod vybraný řádek (M2) provedeme klik na hvězdičku „Add row“ (přidat řádek)

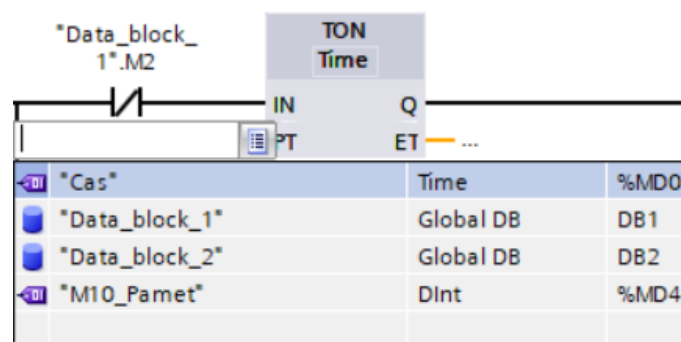
Pokud nevíme jaký datový blok máme použít, najedeme myší na vstup bloku a zobrazí se možné datové typy.



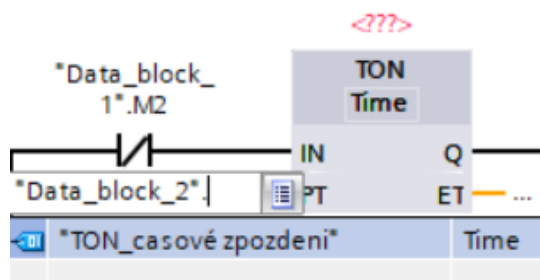
Zapišeme název adresy a vybereme datový typ

M1_Start	Bool	false
M2_Stop	Bool	false
TON_casové zpozdeni	Time	false
Q1_Vystup	RTM	
	Real	
	SInt	
	String	
	Struct	
	Time	
	Time_Of_Day	
	UDInt	

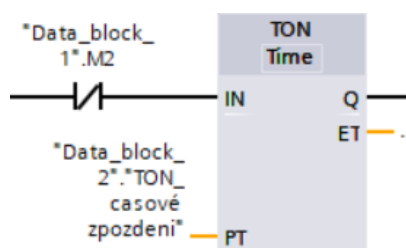
Při přiřazení vstupu nebo výstupu adresy do bloku, musíme zvolit správný blok, do kterého jsme adresu proměnné zapsali. „TON\_casove zpozdeni“ je v „Data\_block\_2“. Klikneme LT na blok



V bloku máme jedinou adresu, kliknutím na ní přiřadíme adresu. Automaticky se zapíše celá adresa: „Data\_block\_2.TON\_casovezpozdeni“.

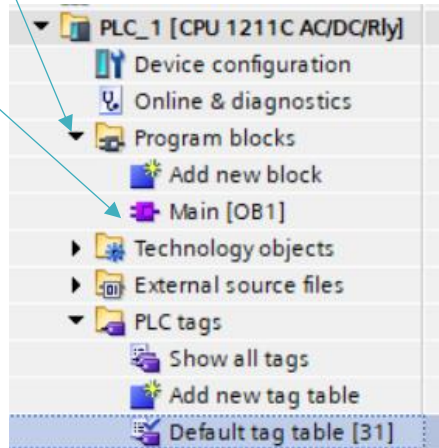


Na obrázku je vidět, že adresa bloku M2 je uložena v datovém bloku „Data\_block\_1“

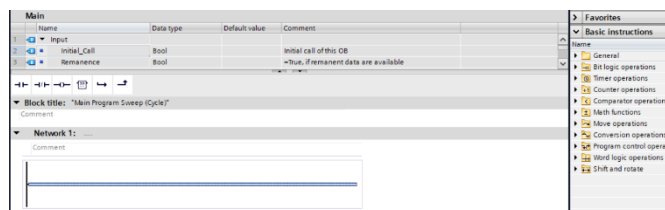


## 6. Otevření programovací plochy

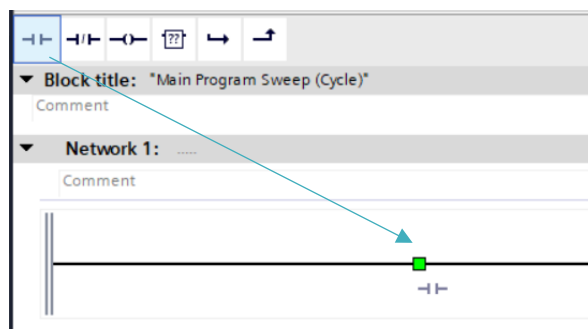
1. Klik na šipku „Program blocks“
2. Klik LT na „Main“ (hlavní)



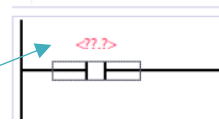
## 7. Programování v LAD



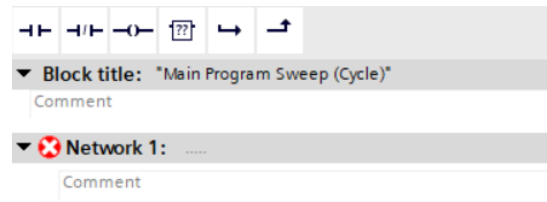
1. Blok uchopíme myší a táhneme na zelený čtvereček na příčce



Po puštění tlačítka blok přiskočí k levé svislé čáře



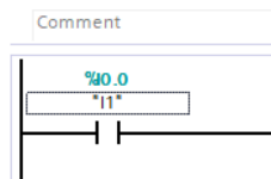
2. Kliknutím na otazníky se otevře okno.



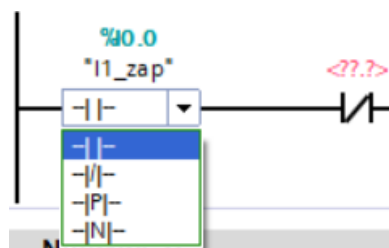
3. Kliknutím na čtvereček v pravé části okna se zobrazí tabulka TAG, zde vybereme řádek, podle vstupu, kterým budeme obvod spínat „I1“.

Symbol	Label	Type	Address
#Initial_Call		Bool	
#Remanence		Bool	
I1	"I1"	Bool	%I0.0
I2	"I2"	Bool	%I0.1
M1	"M1"	Bool	%M0.0
M2	"M2"	Bool	%M0.1
Q1	"Q1"	Bool	%Q0.0
Q2	"Q2"	Bool	%Q0.1

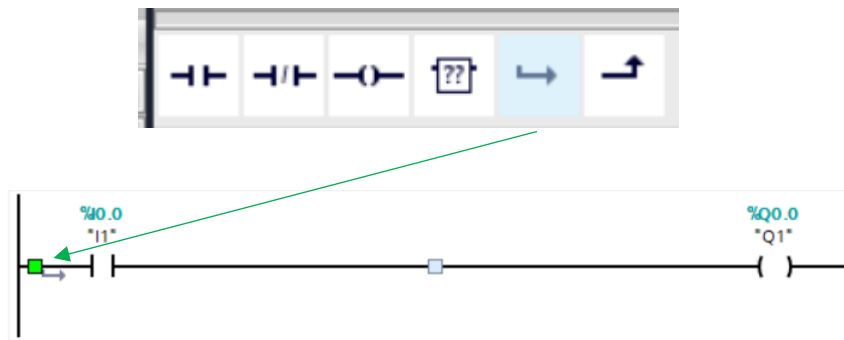
Na kontaktu se zobrazí jeho adresa.



4. Kliknutím do středu kontaktu se otevře okno a kliknutím na rozbalovací šipku a dalším kliknutím můžeme zvolit typ kontaktu: spínací, rozpínací, spínání na vzestupnou hranu „P“ a spínání na sestupnou hranu „N“



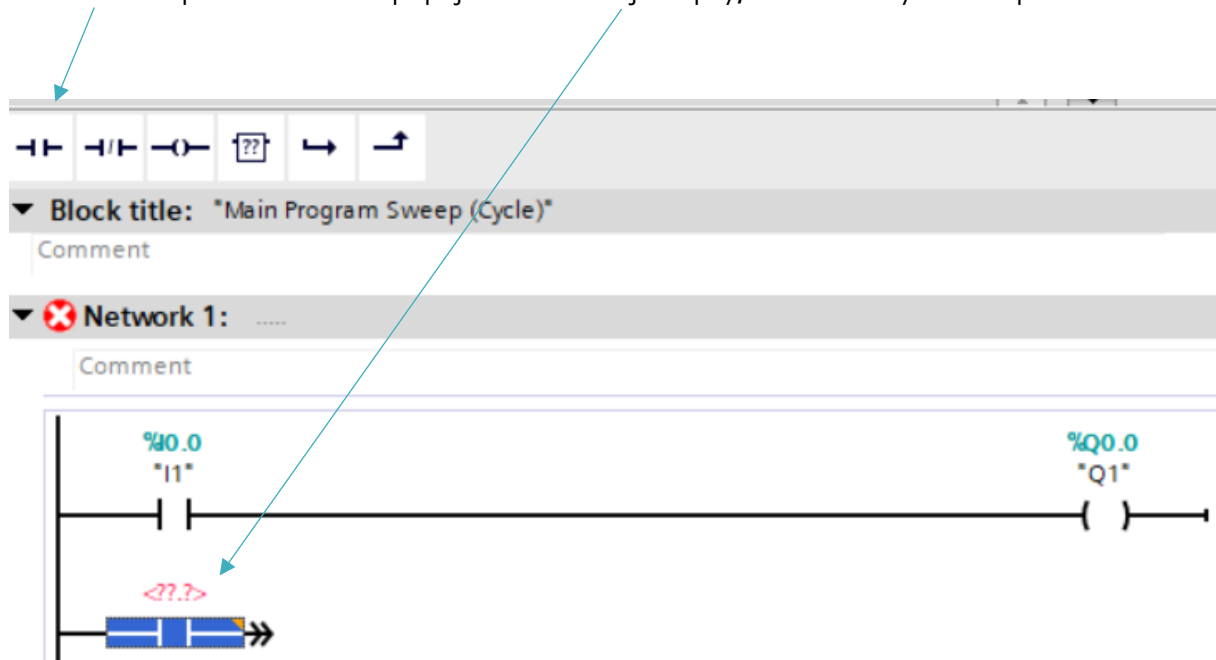
Větvení programu provedeme tažením šipky do bodu větvení (zelený čtvereček)



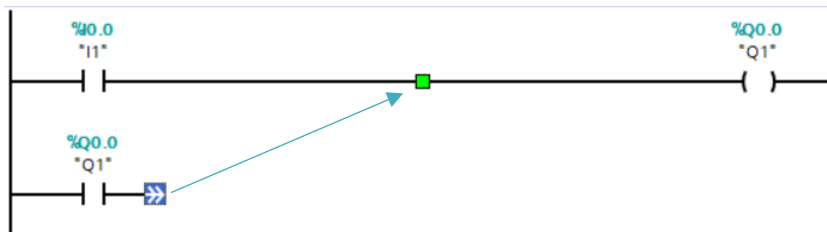
Zobrazí se dvojitá šipka



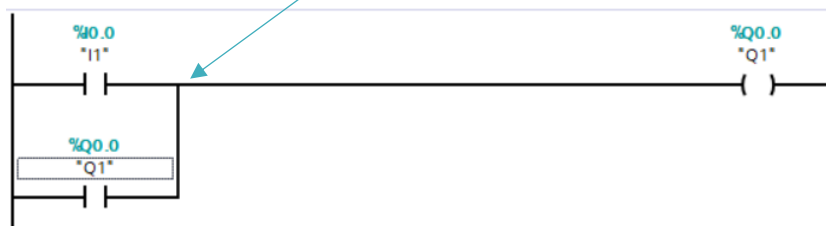
Kliknutím na spínací kontakt se připojí na místo dvojitě šipky, která musí být modře podbarvená



Spojením dvojitě šipky a zeleného čtverečku se provede paralelní připojení vstupu

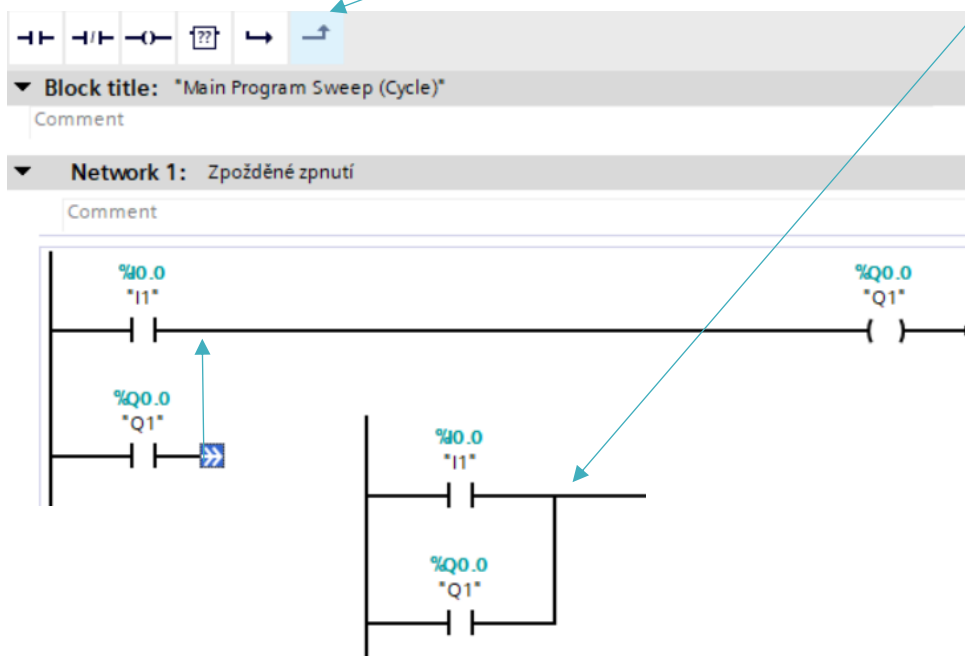


Propojení se automaticky posune k bloku „I1“

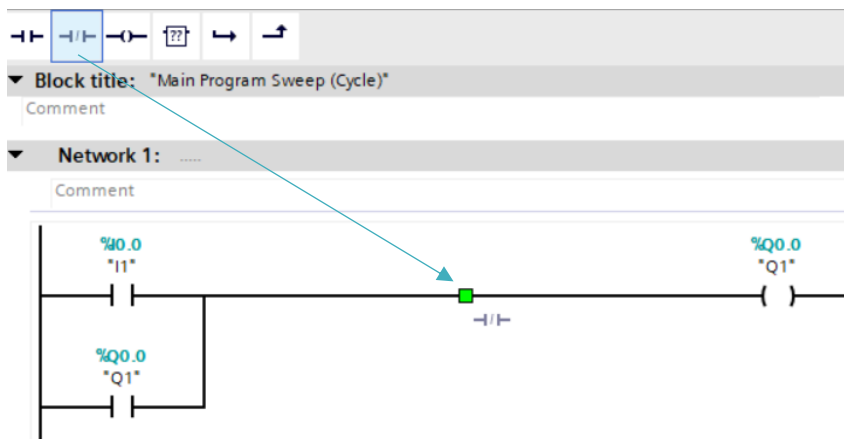


Spínacímu kontaktu přiřadíme adresu „Q1“ to znamená, že se sepne současně s výstupem „Q1“

Pokud je dvojitá šipka modře podbarvená, stačí klik LT na poslední šipku v nabídce a šipka se propojí s prvním řádkem.



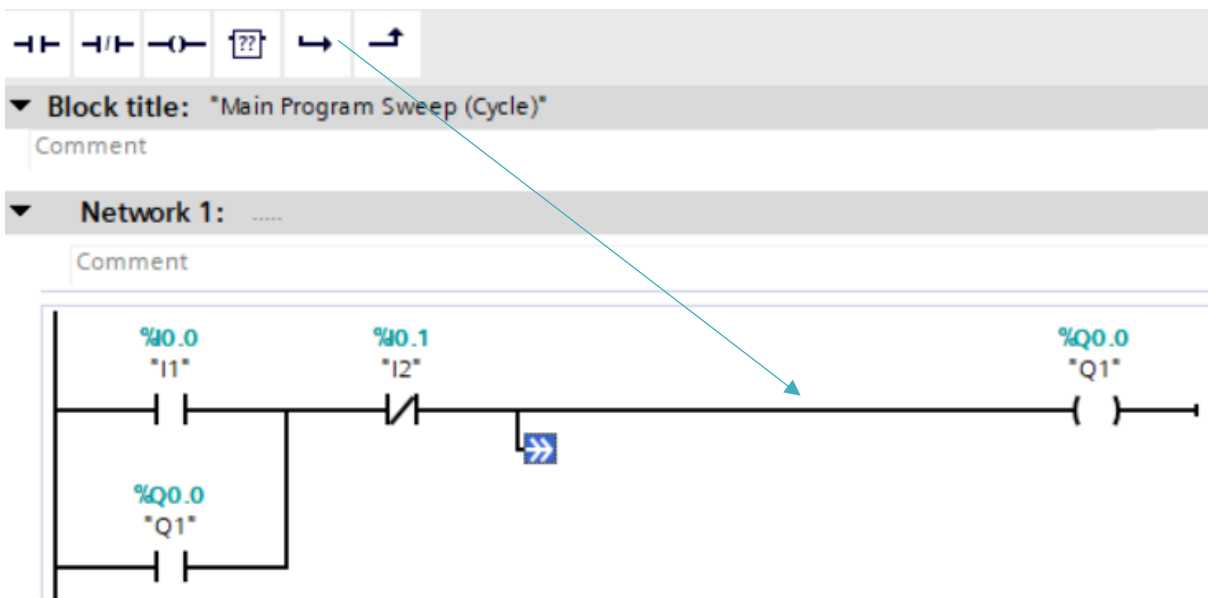
Rozpojovací kontakt vložíme táhnutím bloku na střed hlavní spojovací čáry



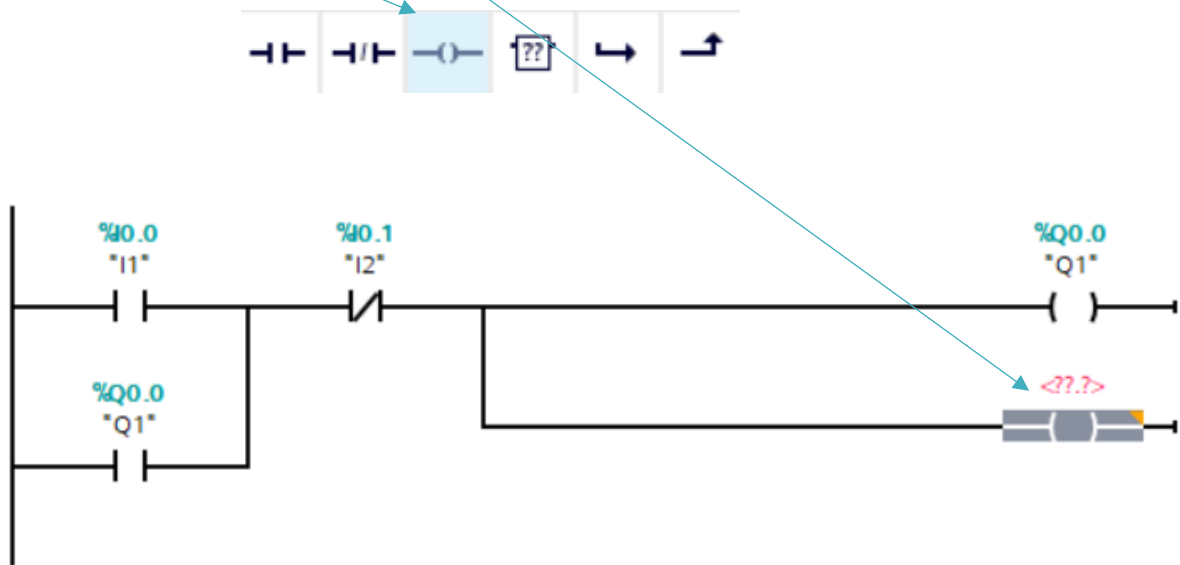
Rozpojovacímu kontaktu přiřadíme adresu „I2“



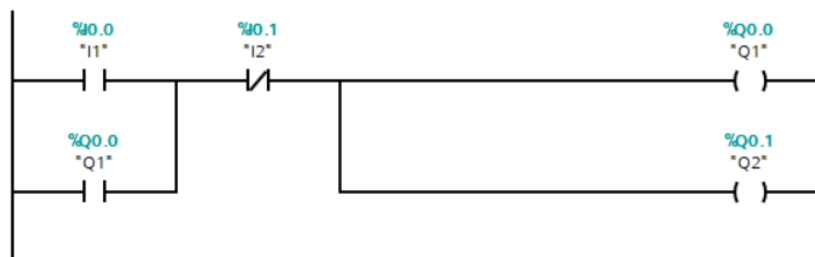
Odbočení za rozpínacím kontaktem provedeme tažením šipky na zelený čtvereček mezi I2 a Q1



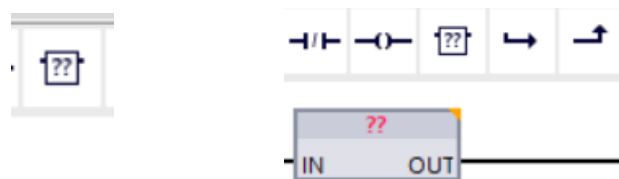
Kliknutím na blok výstupu se blok automaticky připojí k dvojité šipce, pokud je modře podbarvená



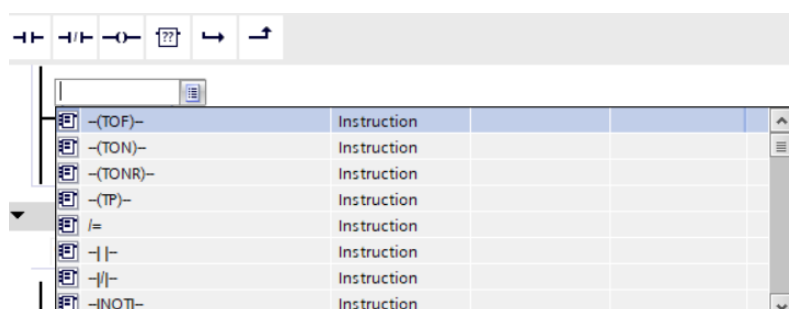
V ukázce vkládání bloků: spínací kontakt, rozpínací kontakt a výstup jsme naprogramovali příklad spínání a vypínání dvou výstupů pomocí dvou tlačítek. „I1“ sepnutí výstupů, „I2“ vypnutí výstupů. Je to klasické zapojení stykače, nebo jinak obvod RS. „Q1“ na spínacím kontaktu představuje pomocný kontakt stykače.



Vložení bloku



Tímto blokem můžeme vkládat celou řadu funkčních bloků z nabídky „Basic Instrukcion“ (Základní instrukce). Kliknutím LT na otazníky se otevře okno, kde dalším kliknutím na rozbalovací čtvereček se zobrazí nabídka abecedně seřazených bloků, z nichž vybereme požadovaný.



## 8. Logické bitové bloky

Bloky pro programování v LAD

Bit logic operations	
	Normally open contact [Shift+F2]
	Normally closed contact [Shift+F3]
	Invert RLO
	Assignment [Shift+F7]
	Negate assignment
	Reset output
	Set output
	Set bit field
	Reset bit field
	Set/reset flip-flop
	Reset/set flip-flop
	Scan operand for positive signal edge
	Scan operand for negative signal edge
	Set operand on positive signal edge
	Set operand on negative signal edge
	Scan RLO for positive signal edge
	Scan RLO for negative signal edge
	Detect positive signal edge
	Detect negative signal edge

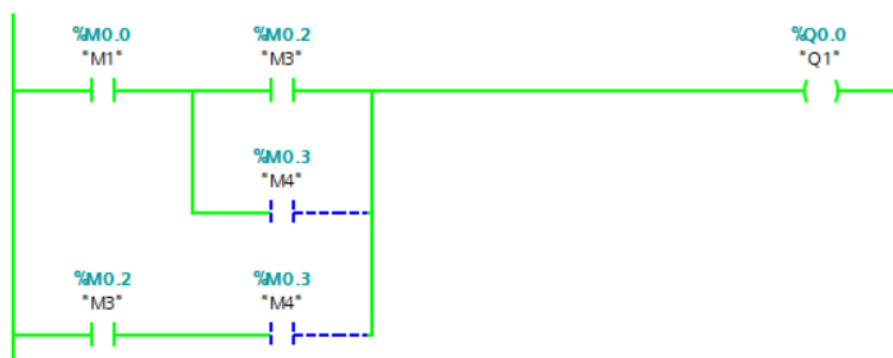
Bloky pro programování v FBD

Bit logic operations	
	AND logic operation [Shift+F2]
	OR logic operation [Shift+F3]
	EXCLUSIVE OR logic operation
	Assignment [Shift+F7]
	Negate assignment
	Reset output
	Set output
	Set bit field
	Reset bit field
	Set/reset flip-flop
	Reset/set flip-flop
	Scan operand for positive sign...
	Scan operand for negative sign...
	Set operand on positive signal ..
	Set operand on negative signal.
	Scan RLO for positive signal ed...
	Scan RLO for negative signal e...
	Detect positive signal edge
	Detect negative signal edge

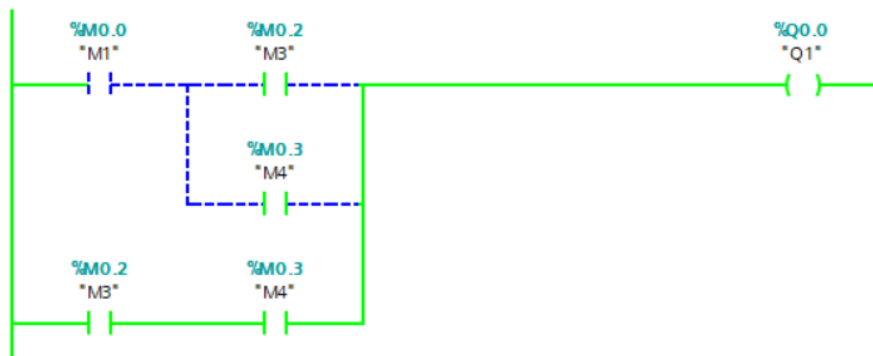
Programování v LAD. Základními prvky jsou spínací kontakt „**Normally open contact**“ (Normálně otevřený kontakt), rozpínací kontakt „**Normally closed contact**“ (Normálně uzavřený kontakt) a **Invert RLO** (invertování – obrácení výsledku logické operace).

Př.: Zapojení logické funkce:  $M_2 (M_3 + M_4) + M_3 + M_4 = Q$

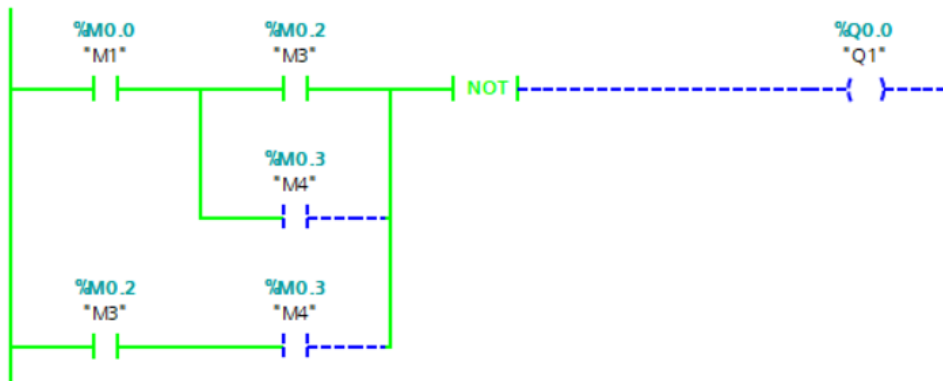
Na obrázku  $M_1$  a  $M_3$  mají hodnotu 1. Všechny kontakty jsou **spínací**



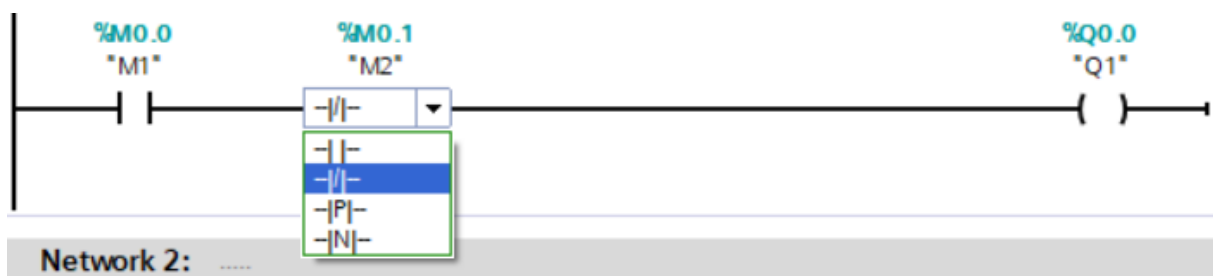
Zde mají hodnotu 1 bloky M3 a M4, výstup Q je sepnutý



Stejný příklad s vloženým blokem **NOT** – negace. Neguje se celý výraz. Je-li M1 a M31 rovno 1, výstup je vypnutý.

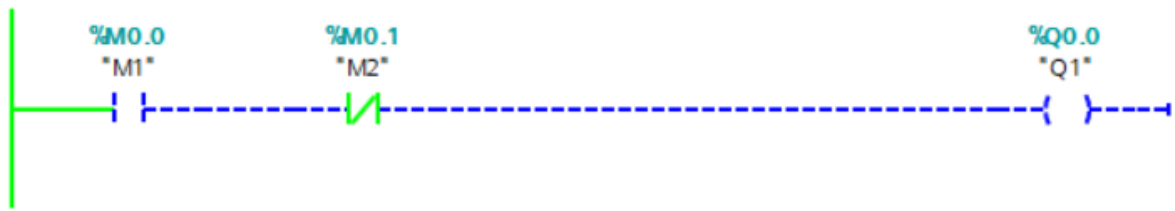


**Rozpínací kontakt.** Máme-li spínací kontakt, můžeme ho změnit na rozpínací kliknutím do středu kontaktu a vybrat po rozvinutí nabídky.



**Rozpínací kontakt** má výstup 1 při vstupu 0 a naopak, je-li vstup 0, výstup Q je 1

M1 = 0, M2 = 0,



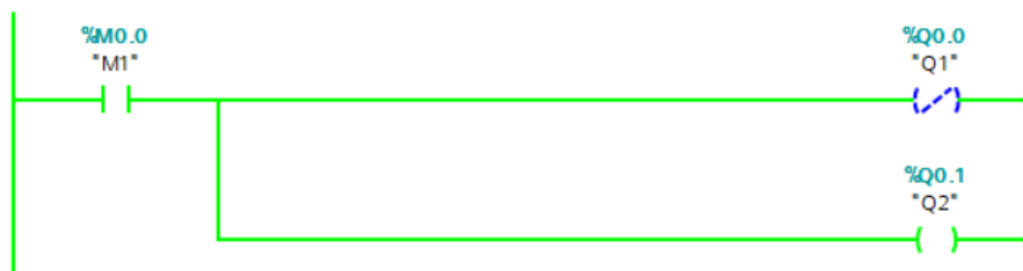
M1 = 1, M2 = 0



M1 = 1, M2 = 1

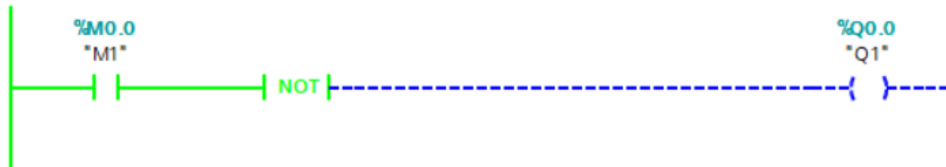


**Výstup** – výstup má tvary sepnutí a vypnutí. Na příkladu je při sepnutém vstupu M1 vypnutý výstup Q1 a sepnutý výstup Q2. Výstup Q1 je negovaný výstup

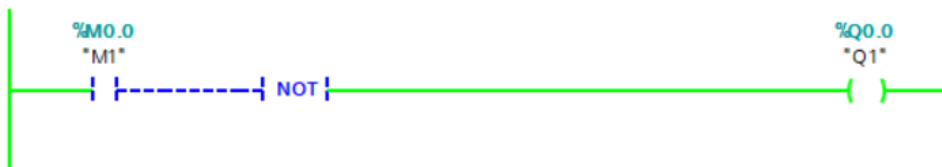


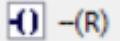
**Negace "Invert RLO"** – blok negace použijete k invertování stavu signálu. Pokud je stav signálu na vstupu instrukce "1", výstup instrukce má stav signálu "0". Pokud je stav signálu na vstupu instrukce "0", výstup má stav signálu "1".

M1 = 1 Q1 = 0




M1 = 0, Q1 = 0



 **R** - „Reset output“ – vypnutí výstupu

Používá se na resetování výstupu, je-li ve stavu 1. Při stavu 0 zůstává stav 0.

 **S** – „Set Output“ – sepnutí výstupu

Pomocí instrukce "Nastavit výstup" můžete nastavit stav signálu zadaného operandu na "1".

Bloky R nebo S můžeme vložit dvěma způsoby.

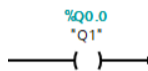
A)

1. Blok "S" natáhneme na programovací plochu z nabídky
2. Klikneme LT na otazníky a vybereme výstup, který budeme spínat

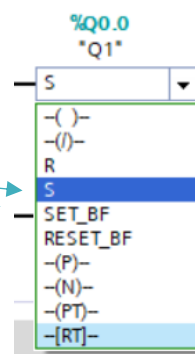
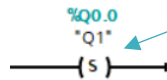


B)

1. Vložíme výstup Q1



2. Kliknutím LT vybereme S



Příklad: Pomocí bloků „S“ a „R“ naprogramujeme obvodu RS pro jeden výstup „Q1“. Tlačítkem „I1“ sepneme výstup „Q1“ a tlačítkem „I2“ výstup vypneme.

1. V zakresleném programu vložíme na výstup dva stejné bloky „Q1“



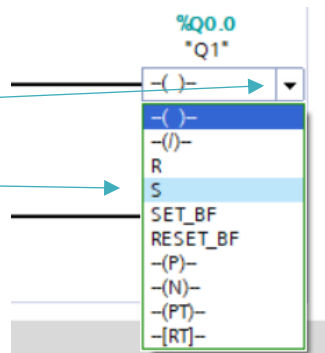
2. Klikneme do prvního bloku „Q1“

3. Klikneme na rozbalovací šipku

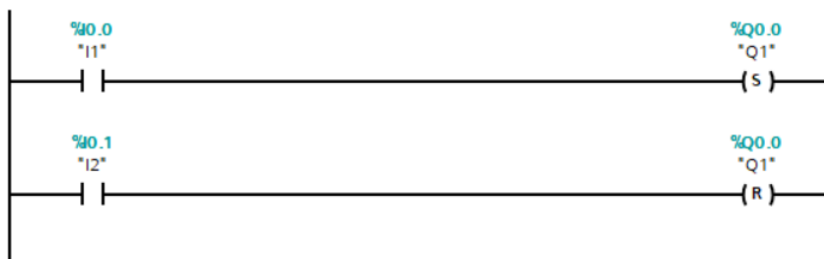
4. Klik LT na S

5. Klik LT do znaku „Q1“ na druhém řádku

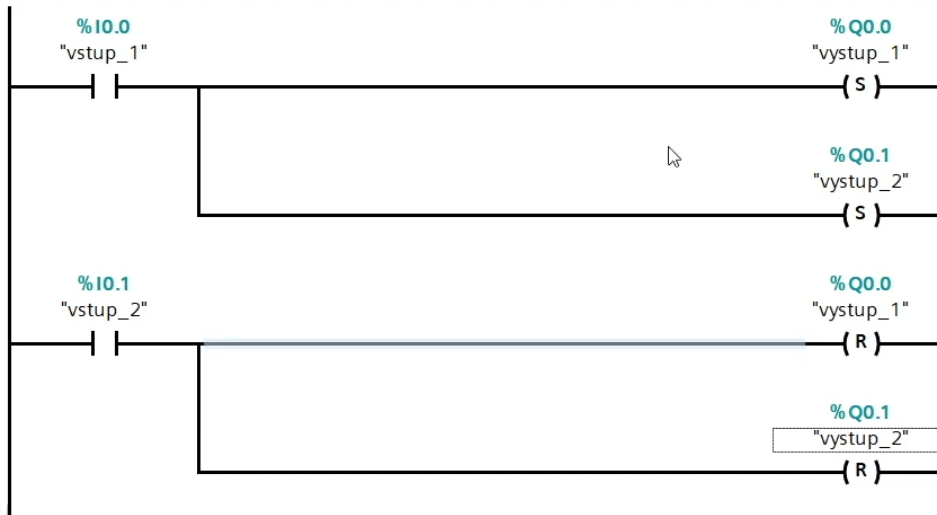
6. Klik LT na „R“

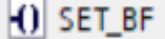


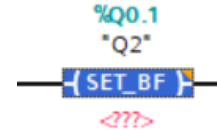
Výsledek: Stiskem tlačítka „I1“ se sepneme výstup „Q1“ a stiskem tlačítka „I2“ se výstup vypne



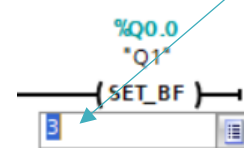
Program pro spínání dvou výstupů jedním tlačítkem a vypínání druhým tlačítkem :



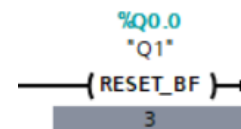
 **SET\_BF** SET\_BF: Set bit field (Nastavení bitového pole)



Blok umožní sepnout počet výstupů zadaných po klik LT na otazníky. Na obrázku je hodnota 3. Tím se najednou sepnou výstupy Q1, Q2, Q3.



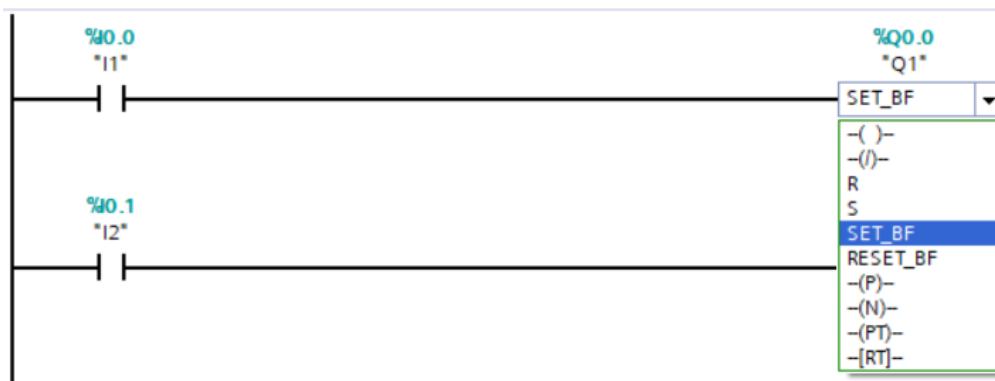
 **RESET\_BF** RESET-BF: Reset bit field (Resetování bitového pole)



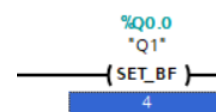
Blok umožní vypnutí počet výstupů zadaných ve spodním poli. Zde je to vypnutí výstupů Q1, Q2, Q3. Zadáme-li 2, vypne se Q1 a Q2.

Příklad programu pro spínání čtyř výstupů jedním tlačítkem a vypínání druhým tlačítkem.

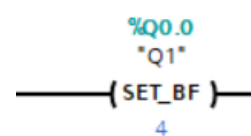
1. Dvojklik LT do bloku Q1 prvního řádku
2. Klik LT na rozbalovací šipku
3. Klik LT na „SET\_BF



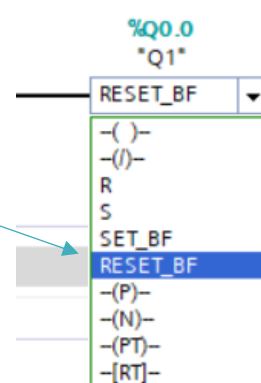
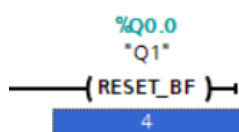
4. Kliknutím pod blok „Q1“ se do okna zapíše počet sepnutých výstupů od Q1 včetně zde 4. Stiskem tlačítka I1 se zapnou první čtyři výstupy.



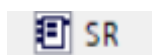
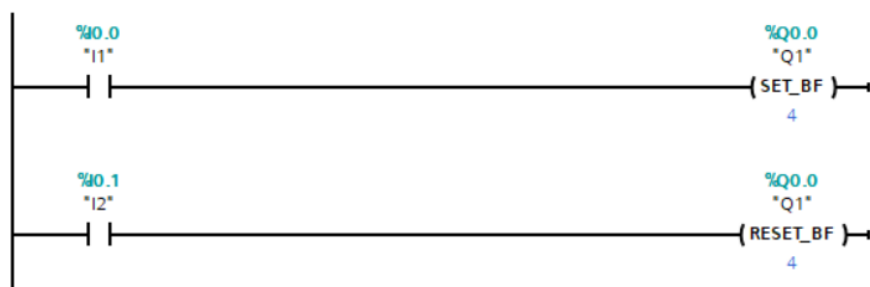
5. Postup se opakuje s výstupem na druhém řádku



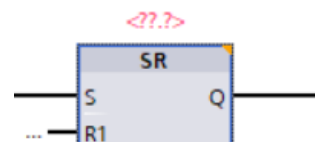
6. Klik na RESET\_FB
7. Kliknutím pod blok „Q1“ se do okna zapíše počet sepnutých výstupů od Q1 včetně zde 4. Stiskem tlačítka I2 se všechny 4 vstupy vypnou



## Výsledný program

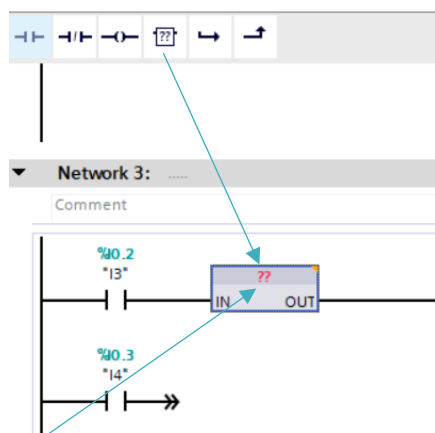


Set/reset flip-flop „Sepnout/vypnout klopný obvod“



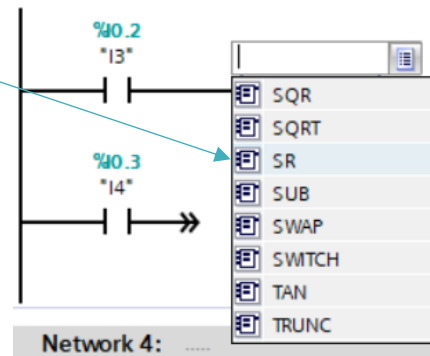
Varianta naprogramování obvodu RS pro jeden výstup „Q1“

1. Blok s otazníky vložíme za vstup „I3“

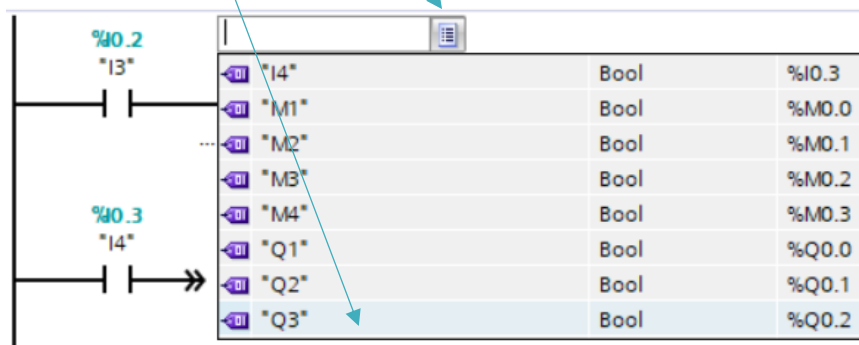
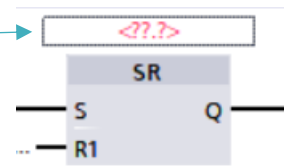


2. Klik LT na otazníky
3. Klik LT na rozbalovací ikonu

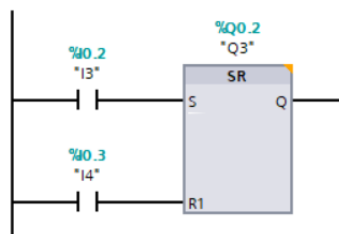
- Klik LT na SR (blok set reset s předností reset- dostane-li se na vstup S i R hodnota 1 výstup vypne (o). U Bloku RS je to opačně.)



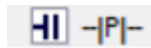
- Klik LT na otazníky
- Klik LT na rozbalovací ikonu
- Označí se požadovaný Výstup „Q3“



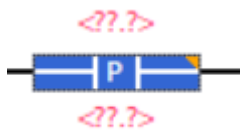
- Propojíme výstup „I4“ se vstupem „R1“



### Kontakt spínání na vzestupnou



hranu. Sepne při stisku tlačítka.



Na místo horních otazníků zadáme adresu spínacího kontaktu (I1). Na místo spodních kontaktů zadáme paměť (M3) do ní se ukládá stav Falze, True (0,1), spojeno rozpojeno.

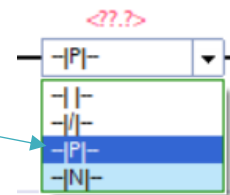
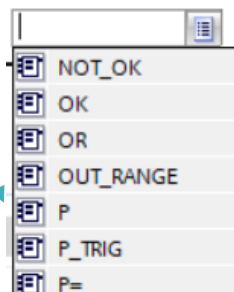
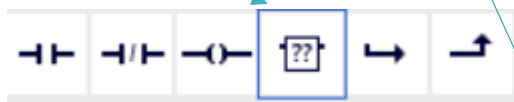
### Kontakt spínání na sestupnou



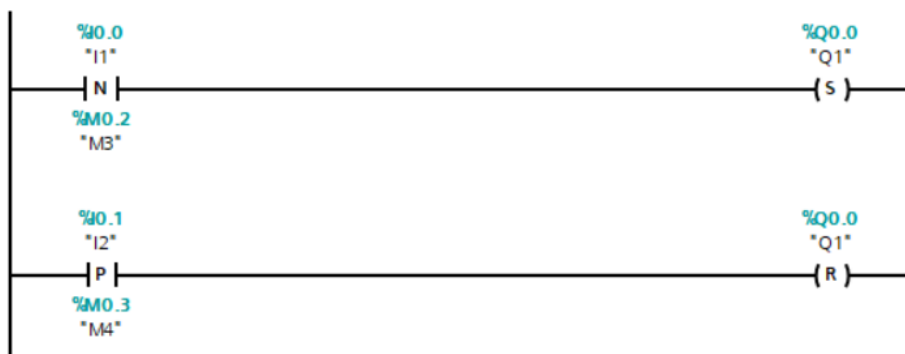
hranu. Sepne při uvolnění tlačítka.

Kontakt je opět možné vložit třemi způsoby.

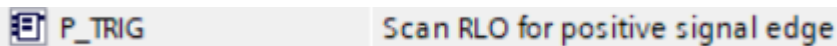
1. vložit spínací kontakt a vybrat stav P
2. natáhnout blok z nabídky Basic instruction
3. vložit obecný blok a najít po rozbalení nabídky



Příklad, kdy po stisku I1 a uvolnění tlačítka se sepne výstup Q1 a při stisku I2 se vypne.

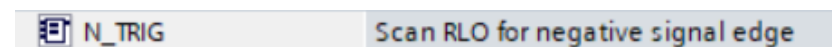


**P\_TRIG** – spínání impulzem na vzestupnou hranu.



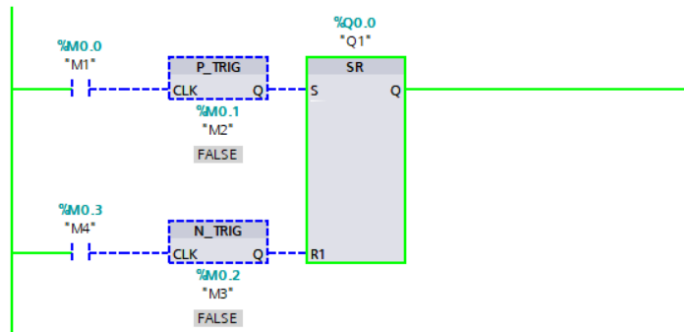
Stiskem tlačítka nebo sepnutím spínače dojde k impulzu na výstupu bloku P\_TRIG.

**N\_TRIG** – spínání impulzem na sestupnou hranu.

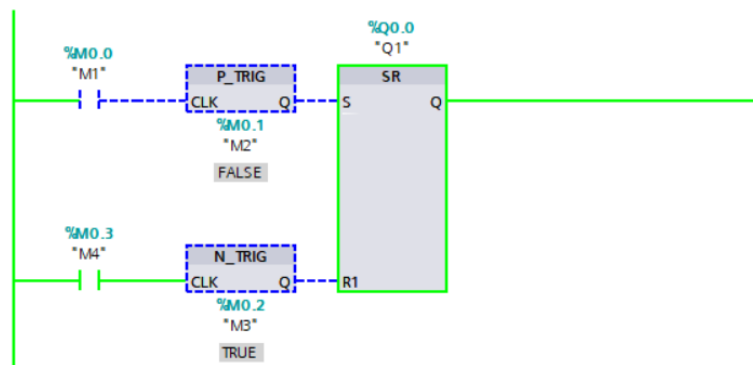


Stiskem tlačítka je na výstupu P\_TRIG hodnota 0, uvolnění tlačítka na výstupu N\_TRIG proběhne impulz 1.

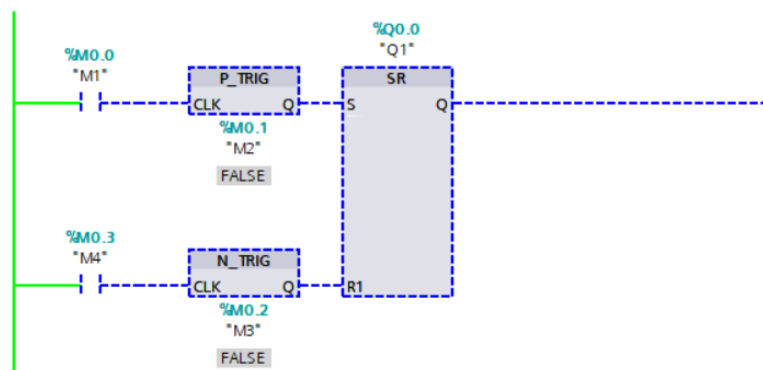
1. Stav kdy bylo sepnuto M<sub>1</sub> a následně vypnuto. Impulzem na vstup bloku SR se výstup bloku sepnul



2. Stav, kdy je sepnuté M<sub>4</sub>, výstup SR je stále sepnutý, vypnutím M<sub>4</sub> dojde k impulzu z N\_TRIG a R<sub>1</sub> vypne výstup SR.



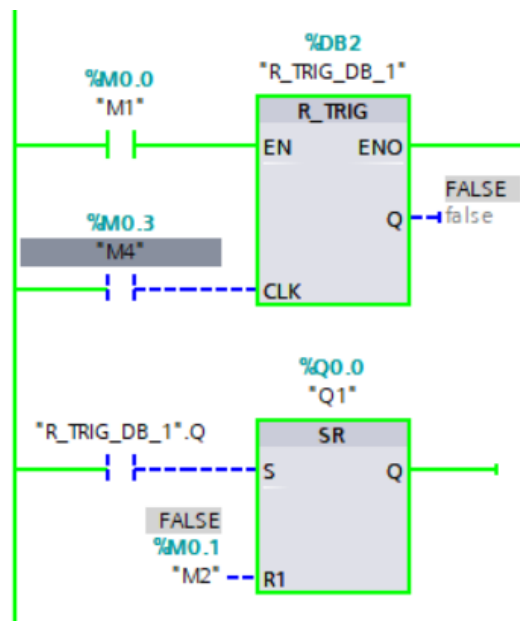
3. Stav po vypnutí M<sub>4</sub>



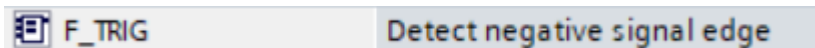
**R\_TRIG** - Detect positive signal edge (Detekce kladné hrany signálu)



Po přivedení hodnoty 1 na vstup CLK dojde na výstupu Q k impulzu 1. Aby bylo možné sledovat, že k impulzu došlo, byl vložen do programu blok SR, který je spínám impulzem. Na obrázku je vidět sepnutí výstupu Q u bloku SR. Vstup CLK bloku R\_TRIG má po uvolnění např. spínacího tlačítka hodnotu 0.



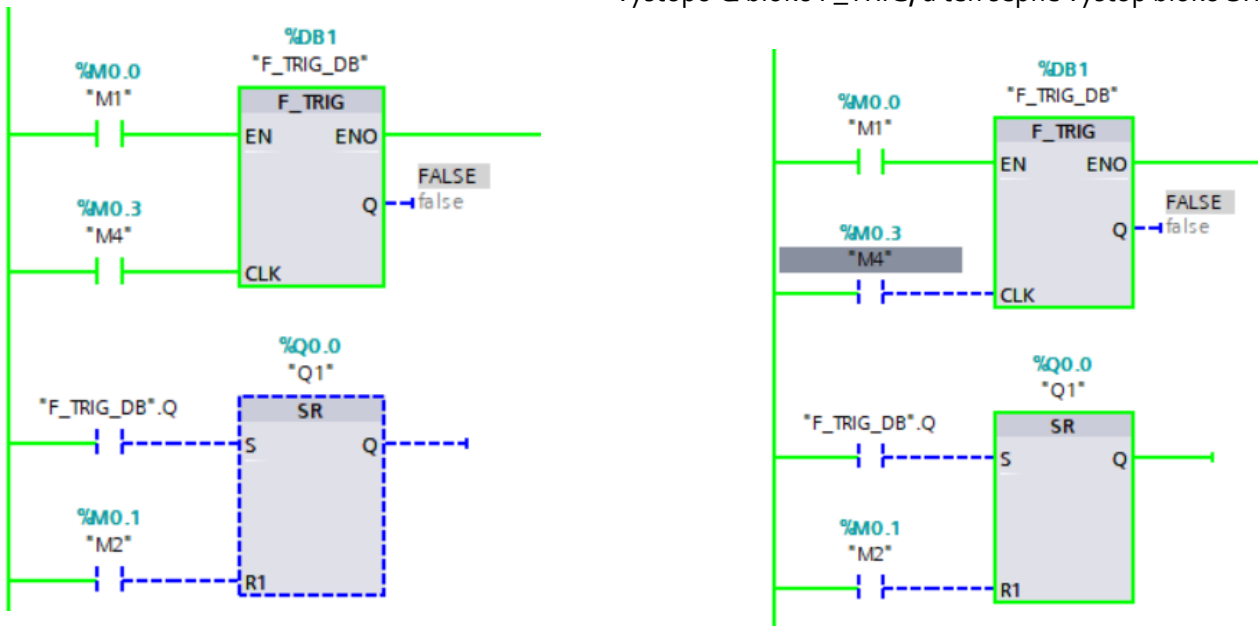
**F\_TRIG**: Detect negative signal edge (Detekce záporné hrany signálu)



Při změně stavu "1" na "0" na vstupu CLK dojde k impulzu na výstupu Q.

Při sepnutí vstupu CLK se nic nestane.

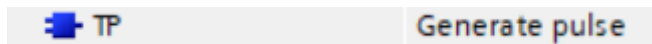
Při vypnutí vstupu CLK dojde k impulzu na výstupu Q bloku F\_TRIG, a ten sepne výstup bloku SR



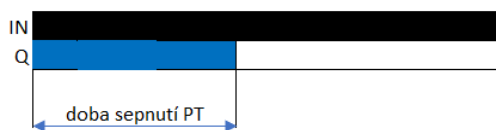
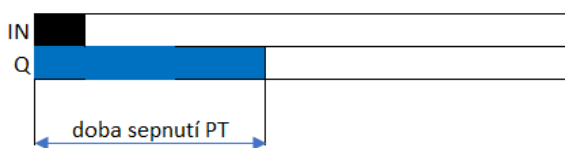
## 9. Časové bloky

Tímer operations	
	Generate pulse
	Generate on-delay
	Generate off-delay
	Time accumulator
	Start pulse timer
	Start on-delay timer
	Start off-delay timer
	Time accumulator
	Reset timer
	Load time duration

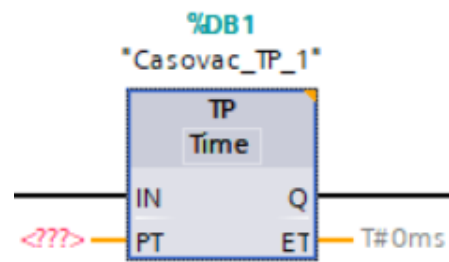
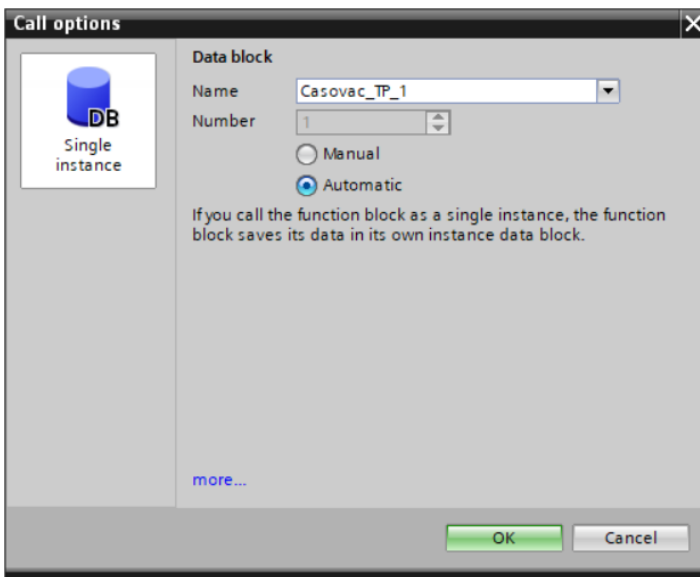
**TP:** Generate pulse (Generovat puls)



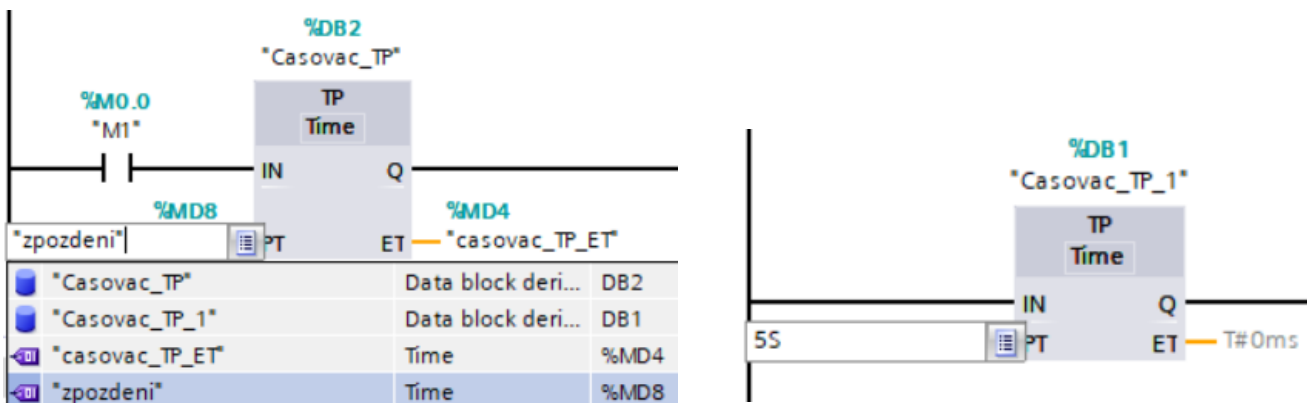
Impulzem nebo trvalým sepnutím vstupu IN se sepne výstup Q na dobu, nastavenou na vstupu PT.



Blok TP natáhneme z nabídky na programovací plochu. Otevře se okno Data block, do kterého napíšeme název bloku. Po té klik na OK. Vloží se blok.

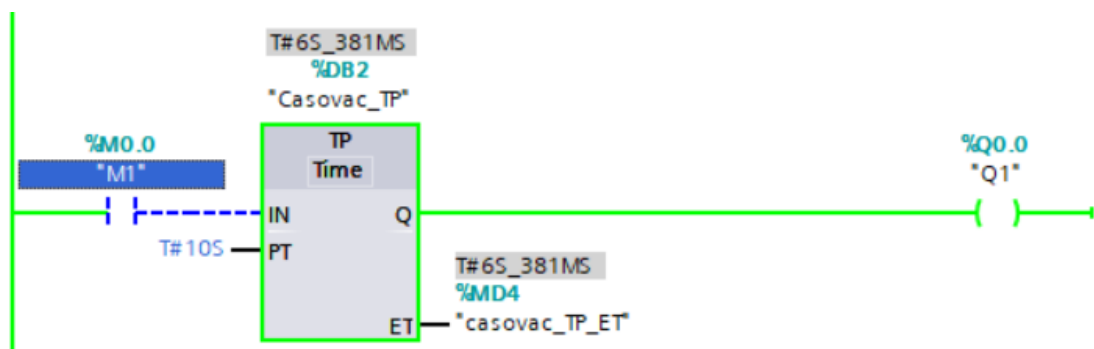


Doba trvání sepnutí výstupu se nastavuje vstupem PT. Kliknutím na otazníky se otevře okno. Do něho se запиše doba sepnutí. Pokud chceme dobu sepnutí např. na panelu HMI klikneme na rozbalovací ikonu a vybereme adresu parametru, který budeme měnit.

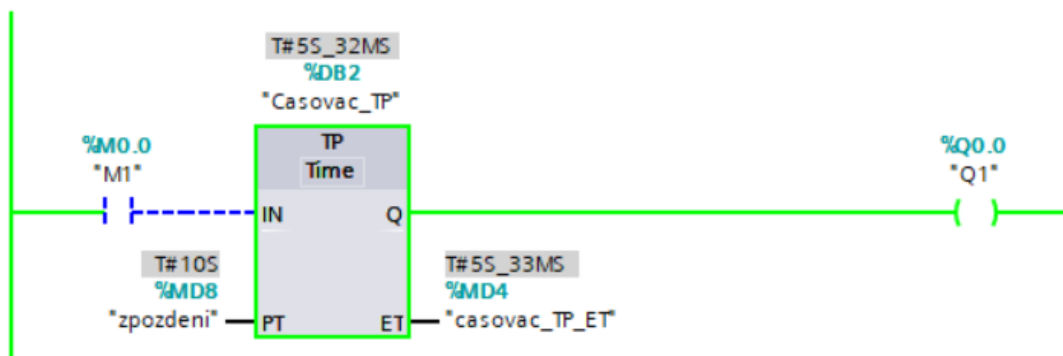


ET je výstup, který můžeme použít např. pro zobrazování na panelu HMI.

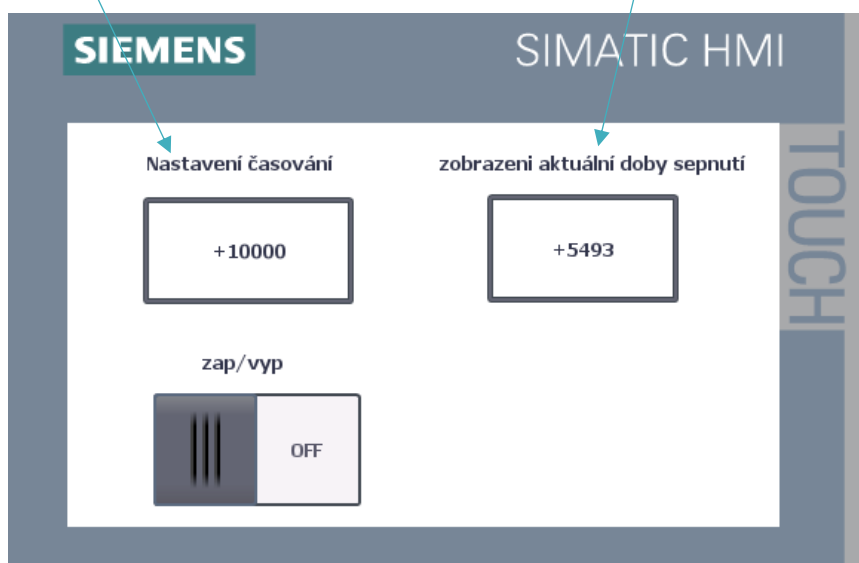
Příklad sepnutí s trvalým nastavením času 10s v programu.



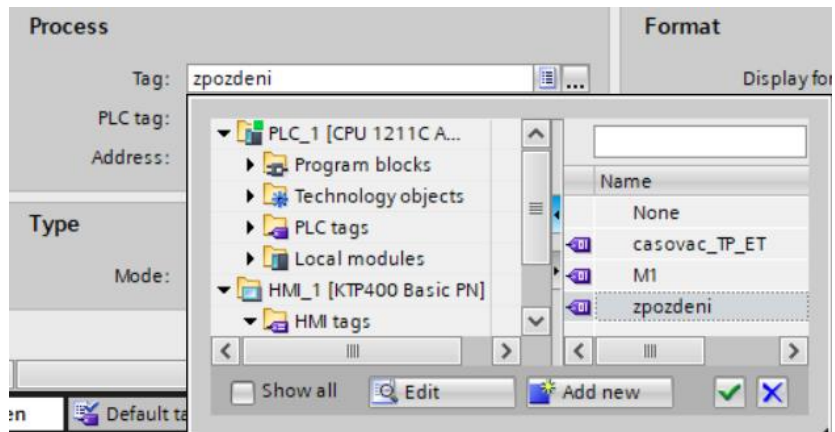
Příklad sepnutí s nastavitelnou dobou na panelu HMI. Čas na panelu HMI je ms.



Na panelu se nastavuje doba sepnutí časování a zobrazuje se aktuální stav časovače.



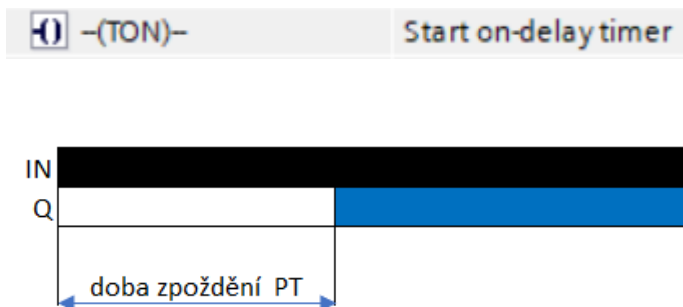
Pro nastavení času se na panelu HMI v bloku zobrazování čísel nastaví adresa PLC tags pro nastavení hodnoty časování a zobrazování.



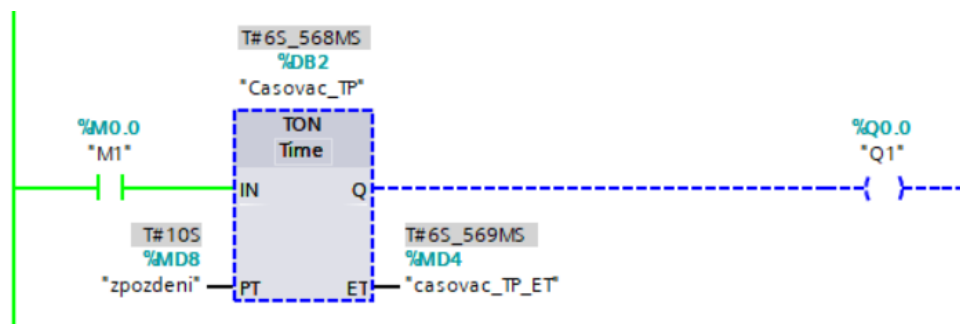
V tabulce tag PC předem zapíšeme adresy vstupu časovače PT a výstupu ET.

I1	Bool	%I0.0
I2	Bool	%I0.1
casovac_TP_ET	Time	%MD4
zpozdeni	Time	%MD8

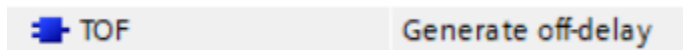
**TON:** Start on-delay timer – zpožděné zapnutí



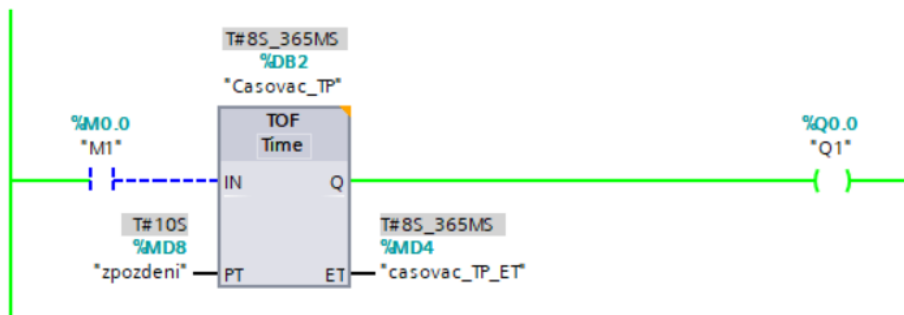
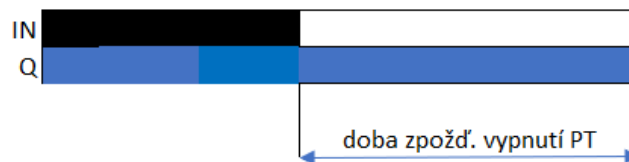
Jako v předchozím případě je možné zadat změnu času PT přímo číslem (např. 5S) na vstupu PT nebo adresou a zadávat ji např. z panelu HMI jak bylo popsáno u časovače TP.



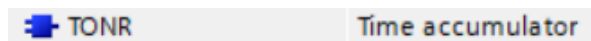
**TOF:** Generate off-delay – zpožděné vypnutí



Po vypnutí IN je výstup Q po nastavenou dobu sepnutý. Po uplynutí nastaveného času se výstup Q vypne.

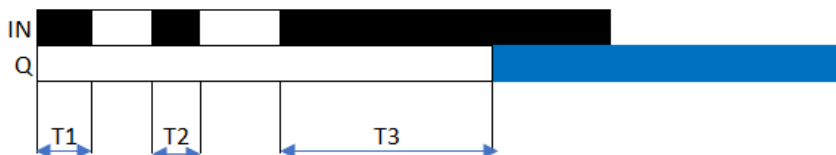


### TONR – Time accumulator

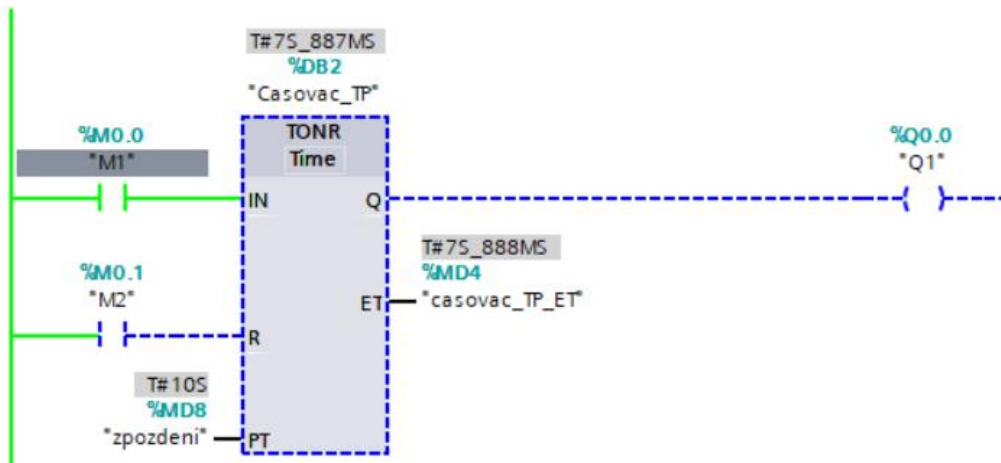


Zpožděné zapnutí s akumulací (časový akumulátor)

Blok TONR složí k akumulaci časových hodnot při přerušovaném vstupu IN. Pokud součet jednotlivých časů sepnutí např.  $T_1 + T_2 + T_3$  dosáhne hodnoty nastavené na vstupu PT, dojde k sepnutí výstupu Q.



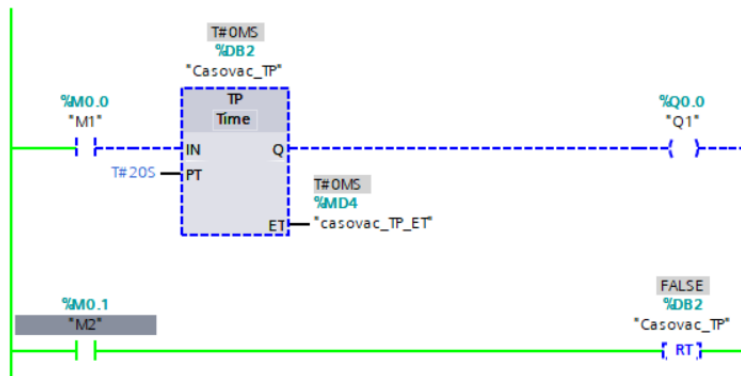
Stav bloku TONR v průběhu časování.



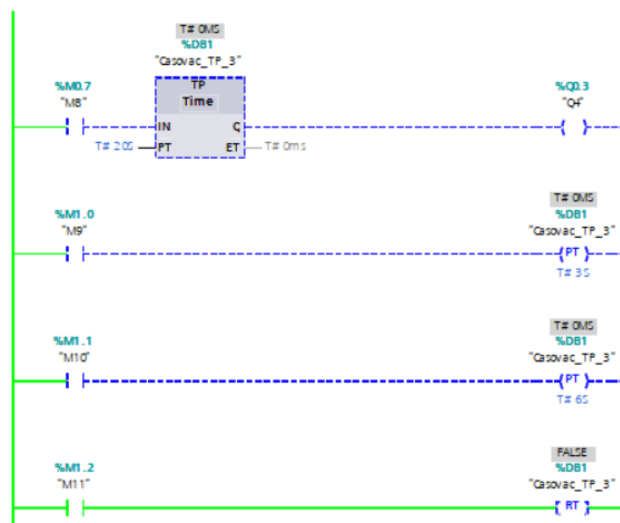
Vypnutí výstupu se provede impulzem na vstup R

**RT: Reset timer** – vypnutí časování

Uvedeme příklad. Je sepnutý časovač „Casovac\_TP“ a jeho výstup. Během časování ho chceme vypnout, K tomu použijeme blok „RT“, který aktivujeme spínačem M2



Příklad zapojení se dvěma bloky PT (pro změnu doby sepnutí časovače) a jednoho bloku RT (pro vypnutí během časování).



PT: Load time duration (doba načítání)

mění dobu na vstupu čítače PT, po sepnutí Bloku PT (IN=1).

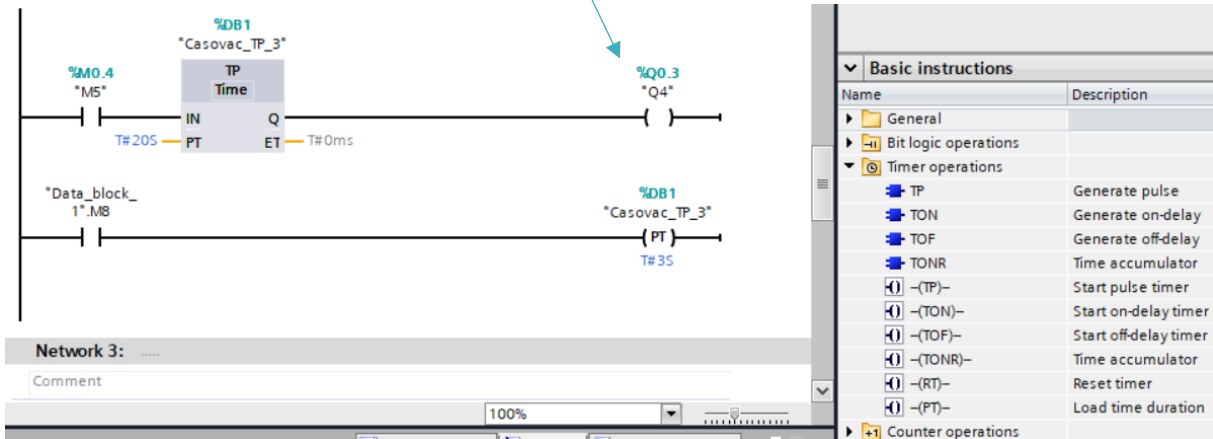
Na uvedeném příkladu je časovač (TP) s nastavenou dobou sepnutí 20s. Do samostatného 2. řádku vložíme spínací kontakt, jeho adresa v příkladu je „Data\_block\_1.M8“ ( adresa může být i z PLC tag).

*Data_block_1*	
M8	Bool
M9	Bool
M10	Bool
M11	Bool

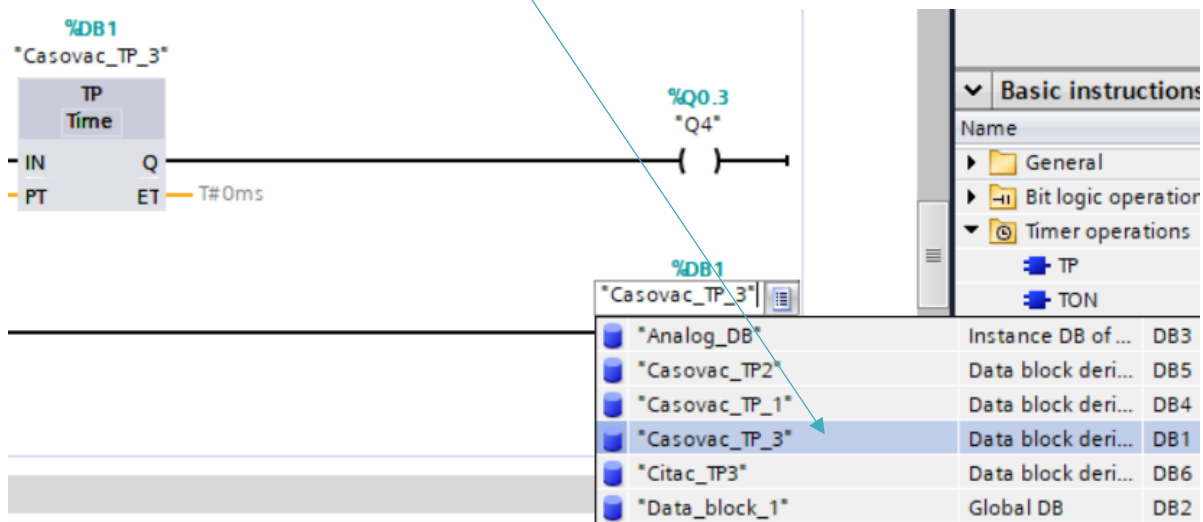
Dále

#Initial_Call	Bool	
#Remanence	Bool	
*Analog_DB*	Instance DB of ...	DB3
*Casovac_TP2*	Data block deri...	DB5
*Casovac_TP_1*	Data block deri...	DB4
*Casovac_TP_3*	Data block deri...	DB1
*Citac_TP3*	Data block deri...	DB6
*Data_block_1*	Global DB	DB2

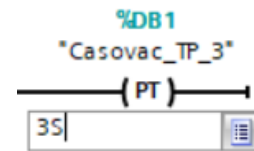
vložíme do druhého řádku blok PT



Adresa bloku PT je adresa „časovače TP\_3“.

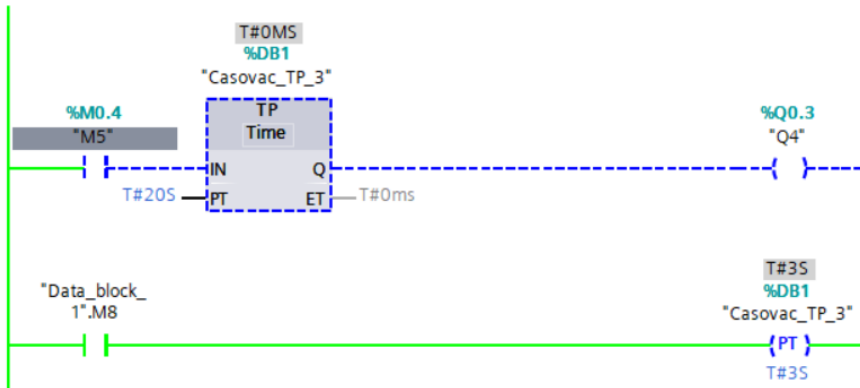


Do spodní části bloku zapíšeme požadovanou dobu sepnutí (3S)



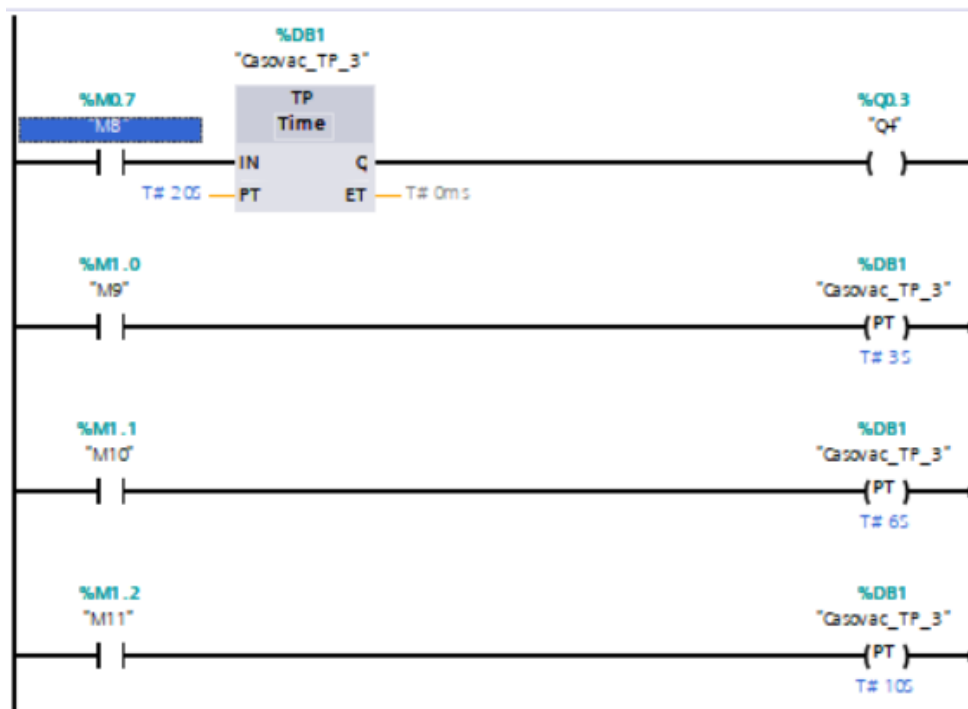
Ovládání:

1. sepneme spínač „Data\_block1.M8“ a následně sepneme spínací kontakt M5. Sepnutí bloku bude trvat 3s, oproti původnímu nastavení 20s.



Zde je příklad kde je možné volit sepnutím jednoho ze čtyř spínacích kontaktů dobu sepnutí výstupu Q časovače „Casovac\_TP\_3“. Je-li sepnutý kontakt M8 (stačí impulz) je doba časování 20s.

Sepneme-li trvale kontakt M9 a k tomu dáme impulz kontaktem M8, do sepnutí bude 3s. Vypneme kontakt M9 a sepneme trvale kontakt M10 a opět dáme impulz spínačem M8, bude doba trvání sepnutí kontaktu 6s. Při opakování tohoto postupu bude sepnutím M11 doba trvání sepnutí Q 10s.

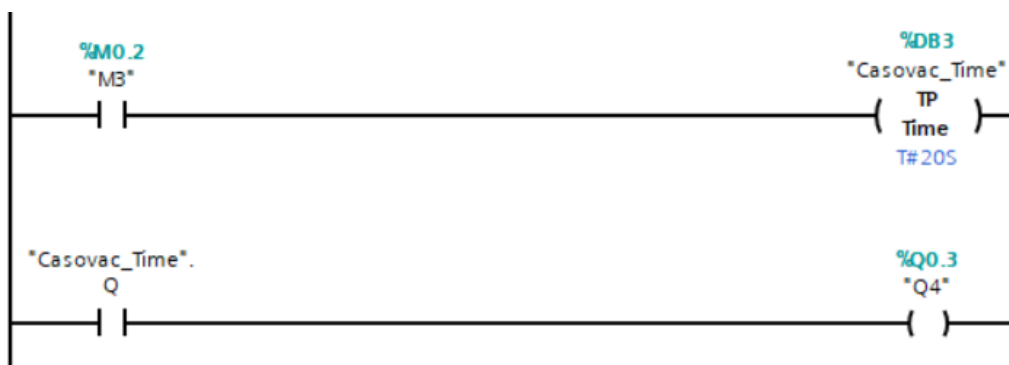


## TP - Start pulse timer



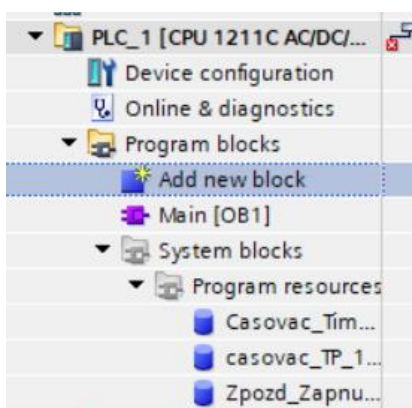
Instrukce „Start pulse timer“ slouží jako časovač, který po impulzu na vstupu je jeho výstup sepnutý na dobu zadanou a to buď pevně, nebo s možností její změny, třeba na panelu HMI.

Zapojení bloků je ve dvou řádkách. V prvním řádku se sepne časování bloku „TP“ a ve druhém se po sepnutí výstupu bloku „TP“ sepne zvolený výstup zde „Q4“.

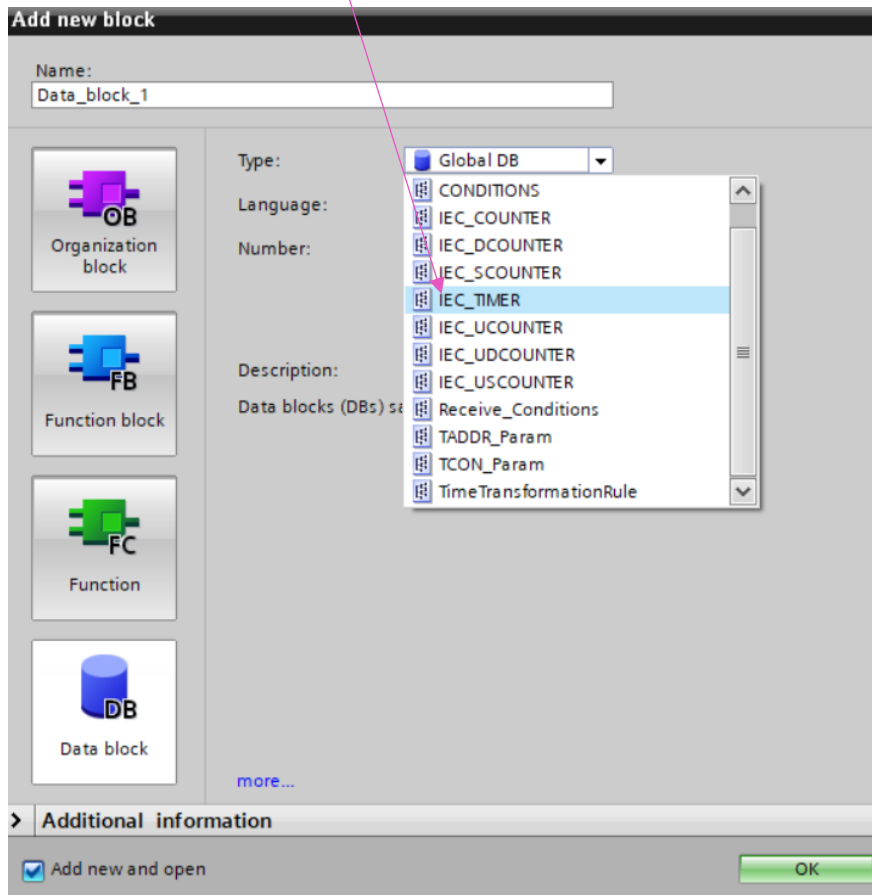


Vložení bloku „TP“

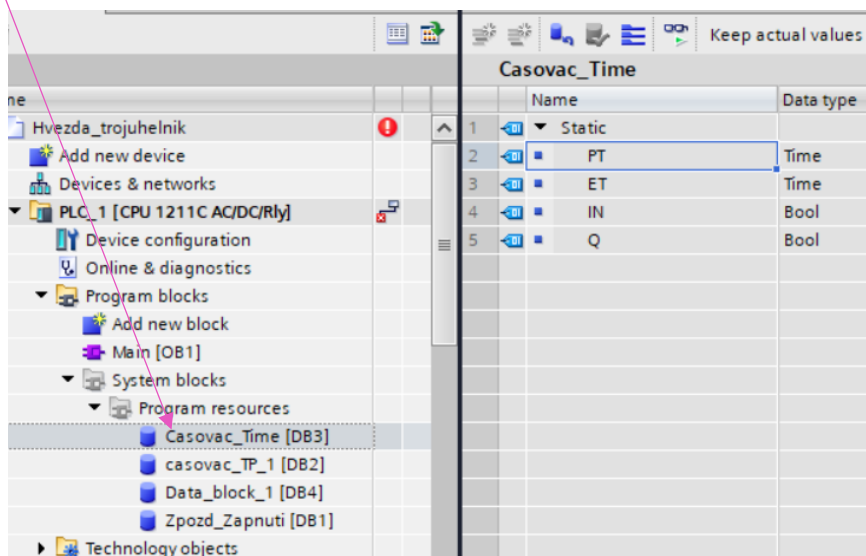
1. Klik LT na „Add new block“



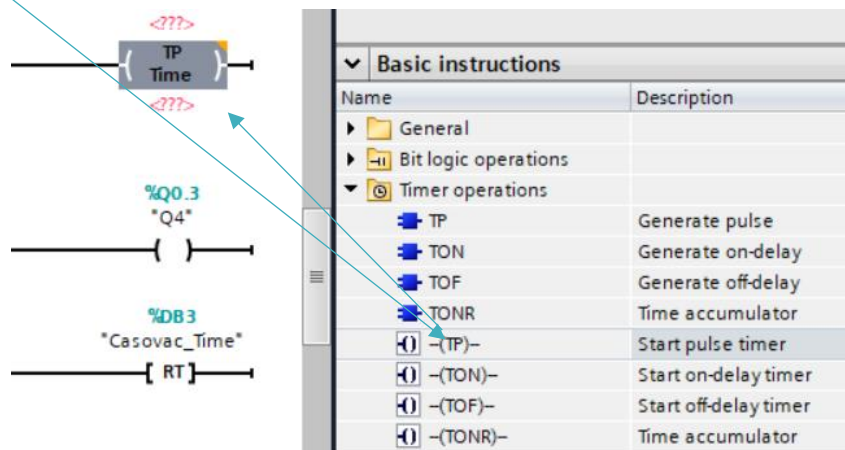
2. v okně Type vybereme „IEC\_TIMER“



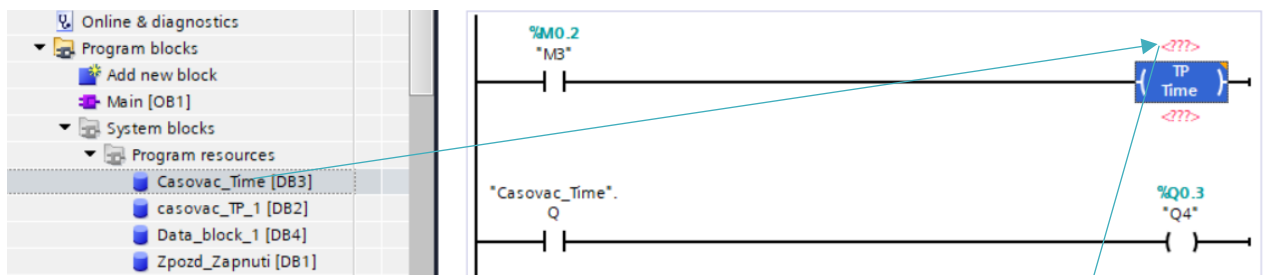
3. OK
4. Ukáže se nový vložený blok v části „Program resources“ (v příkladu je již původní název změněný)



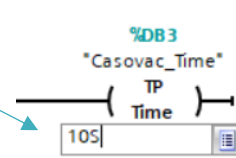
5. Blok TP se natáhne na programovací řádek



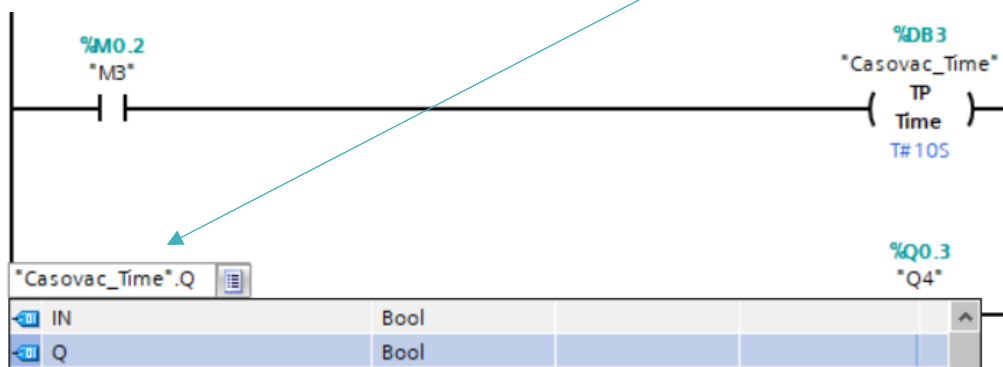
6. „Casovac\_Time (DB3)“ se natáhne na horní otazníky TP.



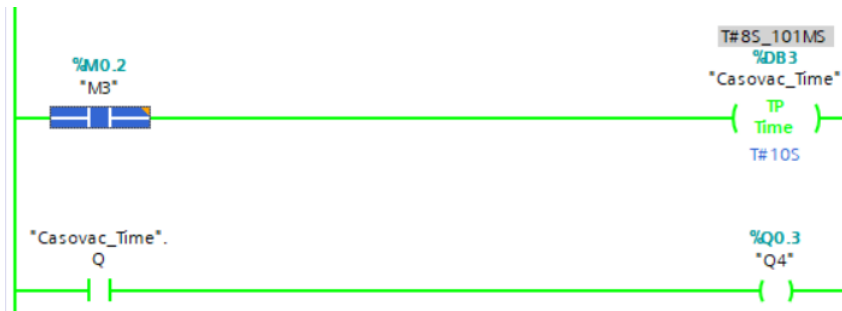
7. Na spodní řádek otazníků se запиše čas sepnutí výstupu



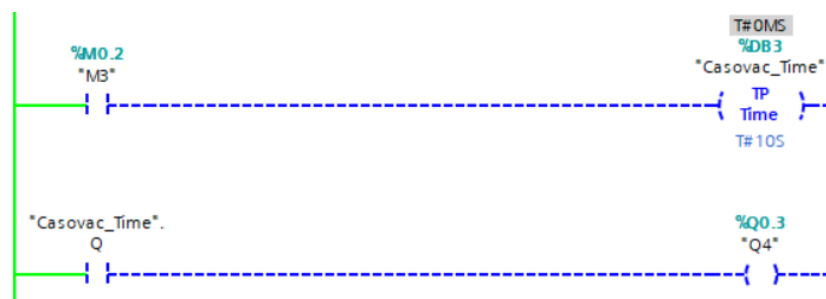
Na spínacím kontaktu na druhém řádku nastavíme adresu časovače TP a jeho výstupu



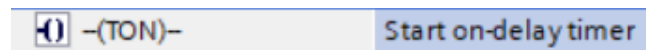
Ukázka programu po sepnutí „M3“.



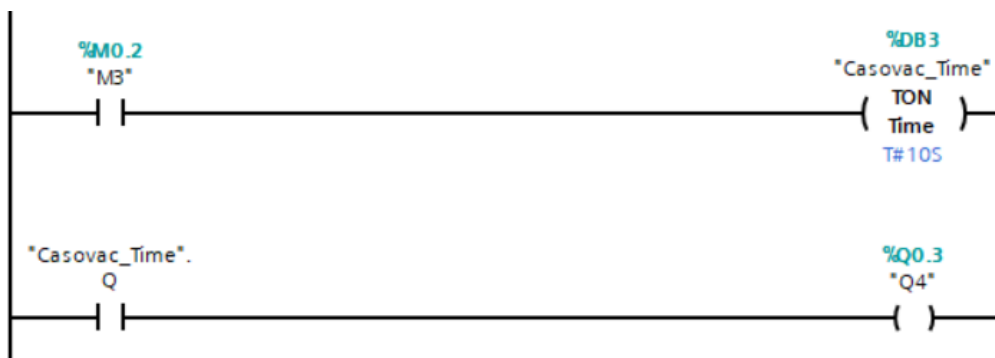
Ukázka programu po dokončení časování



**TON:** Start on-delay timer (zpožděné zapnutí)

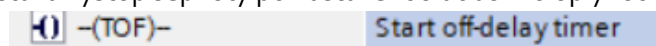


Postup při vkládání bloků je shodný jako u typu TP



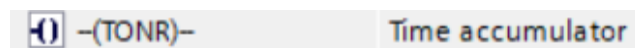
**TOP:** Start off-delay timer (zpožděné vypnutí)

Po vypnutí vstupu, zůstává výstup sepnutý po nastavenou dobu. Po uplynutí doby výstup vypne.

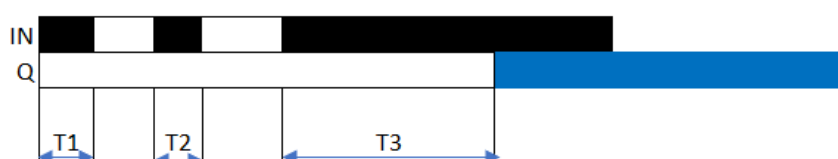


Postup při vkládání bloků je shodný jako u typu TP

**TONR:** Time accumulator (zpožděné zapnutí s akumulací)



Blok TONR složí k akumulaci časových hodnot při přerušovaném vstupu IN. Pokud součet jednotlivých časů sepnutí např.  $T_1 + T_2 + T_3$  dosáhne hodnoty nastavené na vstupu PT, dojde



k sepnutí výstupu Q.

Postup při vkládání bloků je shodný jako u typu TP

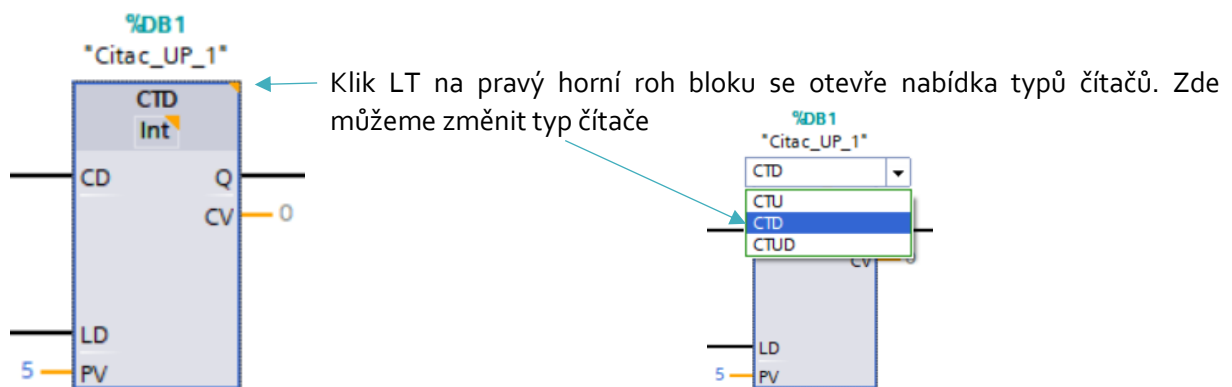
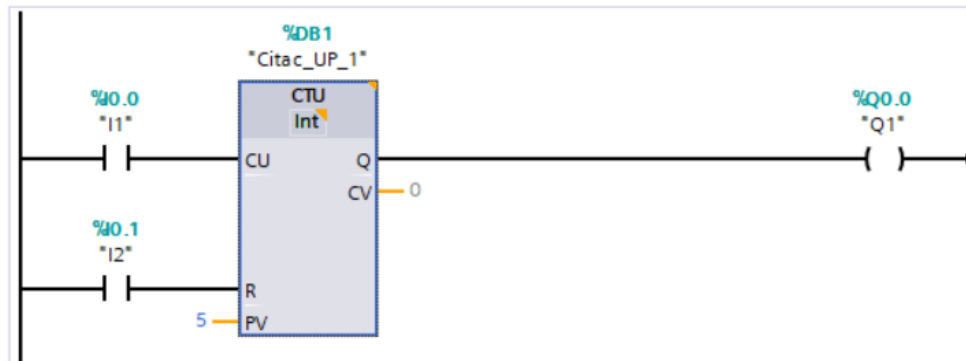
## 10. Čítače

Basic instructions	
Name	Description
General	
Bit logic operations	
Timer operations	
Counter operations	
CTU	Count up
CTD	Count down
CTUD	Count up and down
Comparator operations	
Math functions	
Move operations	
Conversion operations	
Program control operations	
Word logic operations	
Shift and rotate	

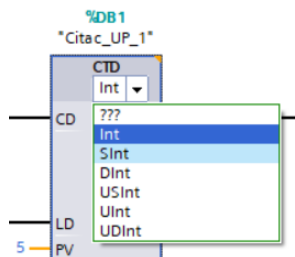
**Čítač CTU** - Zvyšování načítané hodnoty

Při změně na vstupu CU z hodnoty „0“ na hodnotu „1“ se hodnota čítače zvýší o jedničku. Na vstupu PV se nastaví hodnota při které dojde k sepnutí výstupu Q. Přivedením hodnoty „1“ na vstup R se čítač vynuluje.

Př: PV je nastavené na hodnotu 5. Vstup „I1“ budeme zapínat a vypínat. Po pátém zapnutí se sepne výstup Q. Budeme-li dále zapínat a vypínat vstup čítač bude zvyšovat hodnotu a výstup zůstane sepnutý.



Kliknutím LT na pravý horní roh čtverečku uvnitř bloku se dá změnit datový typ – zde je nastaven celočíselný typ Int.



**Čítač CTD – Snižování načítané hodnoty**

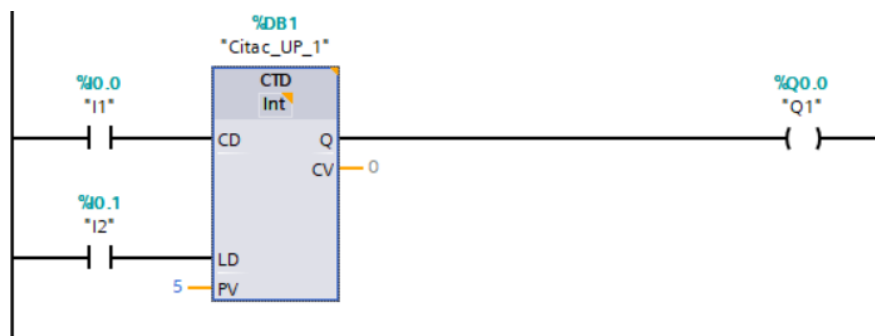
PV – zde se nastaví počáteční hodnota od které budeme odečítat

Vstup CD – spínáním a vypínáním vstupu „I1“ se odečítá hodnota jedna

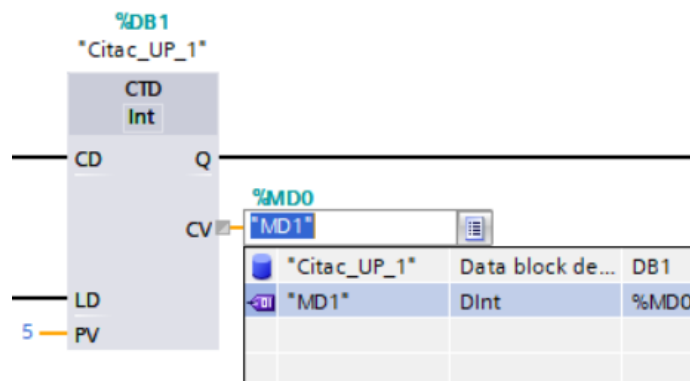
Vstup LD – impulzem na vstup LD se nastaví hodnota čítače na hodnotu zadanou v PV

Při dosažení hodnoty nula se sepne výstup Q.

Př.: Na čítači CTD máme nastavenou hodnotu 5. Sepnutím a vypnutím „I2“ se hodnota 5 nastaví jako výchozí od které se bude odečítat při každém sepnutí a vypnutí „I1“ jednička. Až dosáhne hodnota čítače nuly, tak se sepne výstup. Opětovné nastavení na hodnotu 5 se provede sepnutím a vypnutím vstupu „I2“.



Na výstup CV můžeme zadat adresu paměti do které se bude ukládat hodnota čítače.



Napřed však musíme zadat do tabulky tag zadat adresu. Datový typ je v tomto případě Dint ((Double Integer – dvojitě celé číslo) má délku 32 bitů.

Name	Tag table	Data type	Address
M1	Default tag table	Bool	%M0.0
M2	Default tag table	Bool	%M0.1
Q1	Default tag table	Bool	%Q0.0
Q2	Default tag table	Bool	%Q0.1
Q3	Default tag table	Bool	%Q0.2
Q4	Default tag table	Bool	%Q0.3
M3	Default tag table	Bool	%M0.2
M4	Default tag table	Bool	%M0.3
I1	Default tag table	Bool	%I0.0
I2	Default tag table	Bool	%I0.1
MD1	Default tag table	Dint	%MD0

### Čítač CTDU – Zvyšování i snižování načítané hodnoty

Je to kombinace obou předchozích čítačů.

Vstup CU – zvyšování hodnoty čítače

Vstup CD – snižování hodnoty čítače

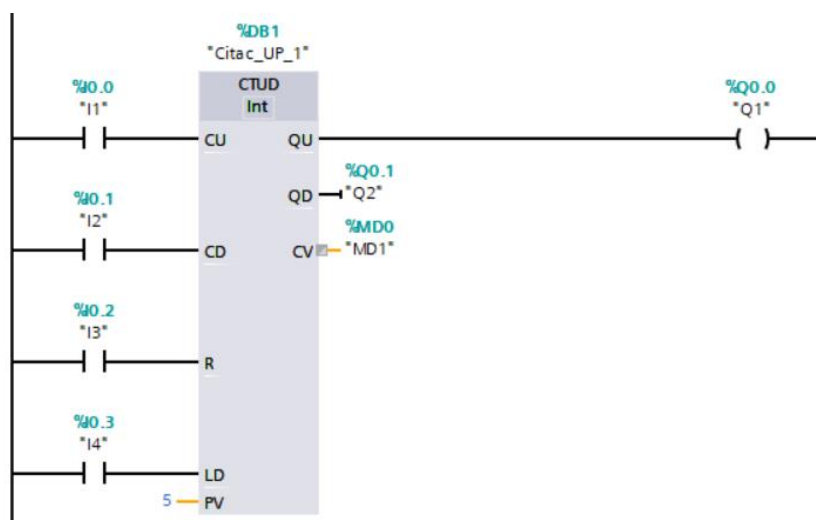
Vstup R – vynulování nastavené hodnoty

Vstup LD – načtení hodnoty PV do čítače

Výstup QU – sepne při hodnotě PV (5)

Výstup QD – sepne při hodnotě PV = 0

CV – paměť aktuální hodnoty



Př.: Výchozí stav bude 0, PV = 5. Impulzy na vstup CU se bude stav čítače o 1 zvyšovat, impulzy na vstup se bude stav čítače o 1 snižovat. Dosáhne-li hodnota čítač 5, sepne se výstup QU a tím i Q1.

Bude-li se hodnota snižovat při dosažení o se vypne výstup QU a sepne se výstup Q2. Impulzem na vstup LD se načte do čítače hodnota 5.

Nemáme-li adresy používaných bloků v tabulce tag, můžeme je kdykoliv přidat.

	I1	Default tag table	Bool	%I0.0
	I2	Default tag table	Bool	%I0.1
	MD1	Default tag table	DInt	%MD0
	I3	Default tag table	Bool	%I0.2
	I4	Default tag table	Bool	%I0.3

## 11. Komparátory

Comparator operations		
	CMP ==	Equal
	CMP <>	Not equal
	CMP >=	Greater or equal
	CMP <=	Less or equal
	CMP >	Greater than
	CMP <	Less than
	IN_Range	Value within range
	OUT_Range	Value outside range
	- OK -	Check validity
	- NOT_OK -	Check invalidity

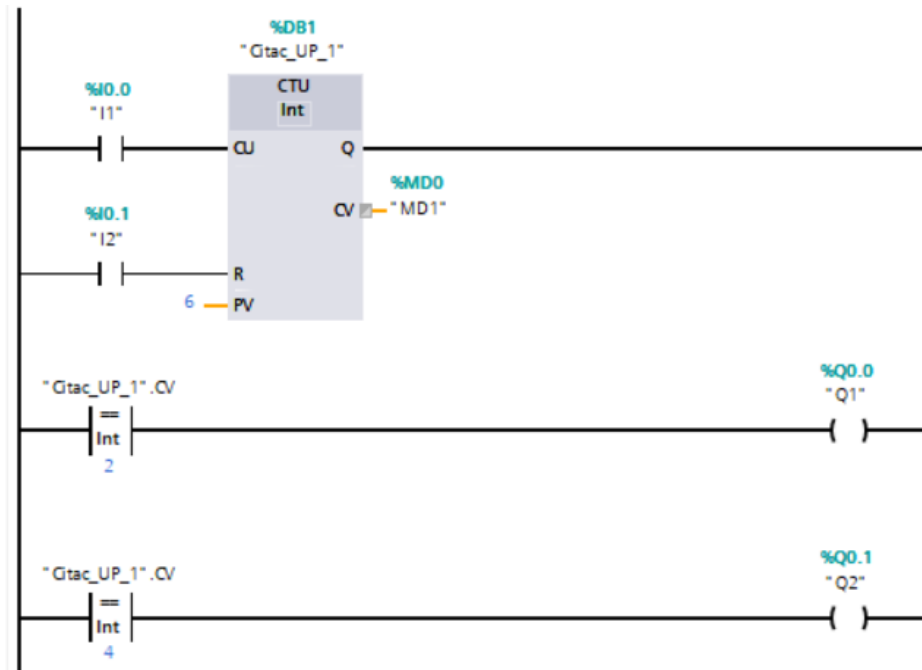
CMP == Equal (rovná se)

V následujícím programu je zapojen čítač a dva komparátory CMP. Jeden s výstupem Q1 a druhý s výstupem Q2. Komparátor 1 má nastavenou hodnotu sepnutí 2 a Komparátor 2 má hodnotu sepnutí 4. Čítač načítá impulzy na vstup CU. Po dosažení hodnoty 2 se sepne výstup Q1. Při zvýšení na hodnotu 3 se vypne, při hodnotě 4 se zapne výstup Q2 a při hodnotě 5 se vypne.

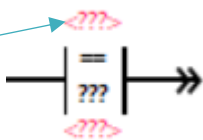
Stav výstupů uvedeného příkladu při různých hodnotách čítače

hodnota čítače	1	2	3	4	5
výstup Q1	OFF	<b>ON</b>	OFF	OFF	OFF
výstup Q2	OFF	OFF	OFF	<b>ON</b>	OFF

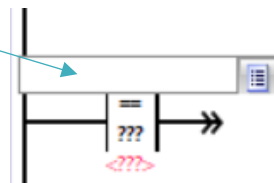
Program:



Důležité je nastavit bloky komparátoru:

10. Klik LT na otazník nad  blokem

11. Klikneme do okna



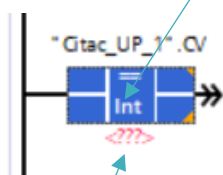
12. Vybereme adresu čítače, nezobrazí-li se nabídka napíšeme první písmena názvu adresy čítače.

*Citac_UP_1*	Data block d...	DB1
*IEC_Counter...	Data block d...	DB2
*Komparator...	Int	%MWD
*Komparator...	Int	%MWD4
*MD1*	DInt	%MDO

13. Klik LT na rozbalovacího tlačítka a vybere se index CV (hodnota výstupu čítače).

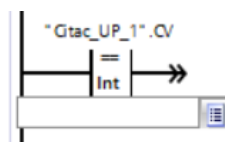
"Citac_UP_1".	
CV	Int
PV	Int

14. Klik LT do středu bloku a vybereme datový typ Int (celé číslo)



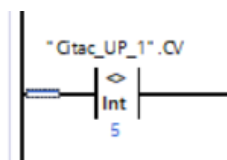
15. Klik LT na spodní otazníky

16. Do okna se zapíše hodnota, kdy má



výstup sepnou

**CMP** – Not equal (nerovná se)



CMP – výstup je sepnutý je-li hodnota čítače různá od nastavené hodnoty zde 5. Při hodnotě 5 je výstup vypnutý.

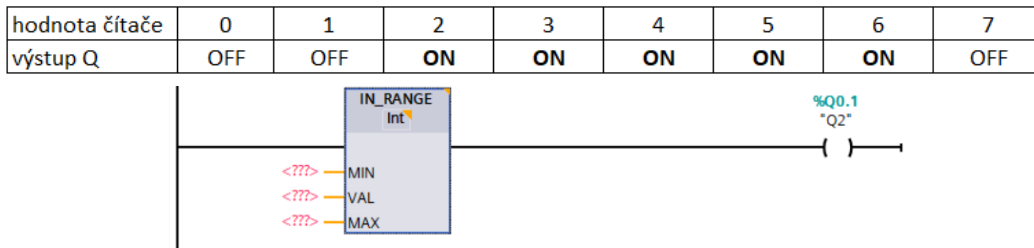
**CMP ≥** Greater or equal (větší nebo rovné)

**CMP ≤** Less or equal (menší nebo rovné)

**CMP >** Greater than (větší než)

**CMP <** Less than (menší než)

## IN\_RANGE: Value within range (hodnota v rozsahu)



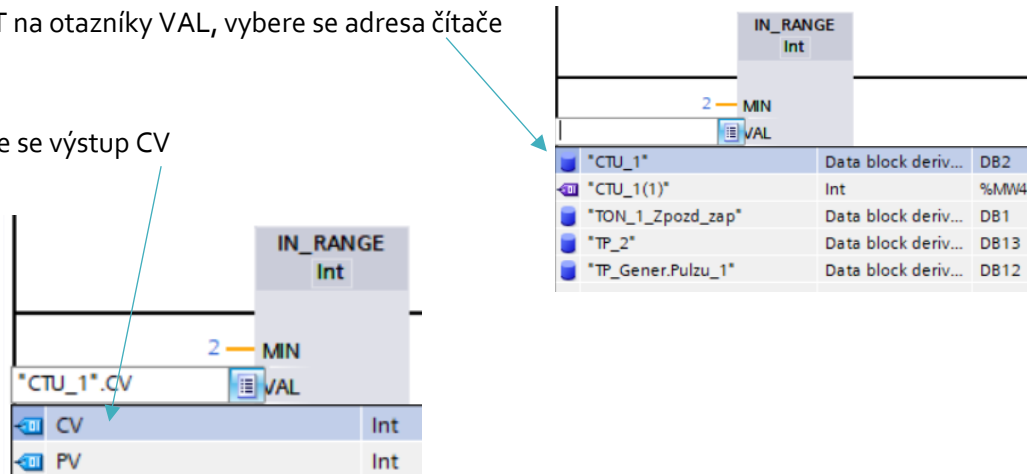
**MIN** – Minimální hodnota pro sepnutí (v příkladu je nastavena na 2 – při dosažení této hodnoty výstup sepne)

**MAX** – Maximální hodnota, nad tuto hodnotu se výstup vypne ( v příkladu nastavená je hodnota 6, zvýší-li se na 7 výstup vypne)

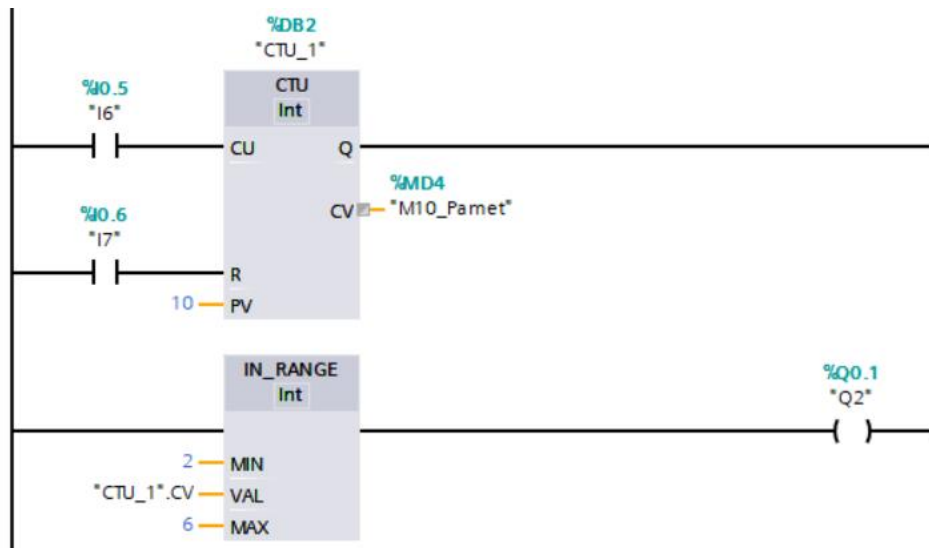
**Val** – adresa výstupu, kde se zaznamenává hodnota, zde je to čítač. Je to adresa čítače a jeho výstupu CV.

1. Klik LT na otazníky VAL, vybere se adresa čítače

2. Vybere se výstup CV



## Celkové zapojení bloků















**OUT\_RANGE:** Value outside range (Hodnota mimo rozsah)

Zde je výstup sepnutý pokud je hodnota vstupu VAL, v tomto případě čítače mimo rozsah MIN – MAX.

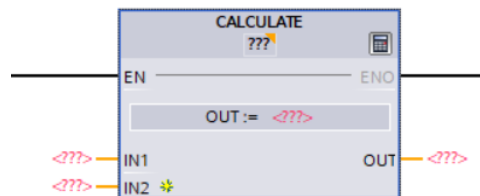
hodnota čítače	0	1	2	3	4	5	6	7
výstup Q	<b>ON</b>	<b>ON</b>	OFF	OFF	OFF	OFF	OFF	<b>ON</b>

## 12. Matematické funkce

Nabídka matematických funkcí

	SQR	Form square	
	SQRT	Form square root	
	LN	Form natural logarithm	
	EXP	Form exponential value	
	SIN	Form sine value	
	COS	Form cosine value	
	TAN	Form tangent value	
	ASIN	Form arcsine value	
	ACOS	Form arccosine value	
	ATAN	Form arctangent value	
	FRAC	Return fraction	
	EXPT	Exponentiate	

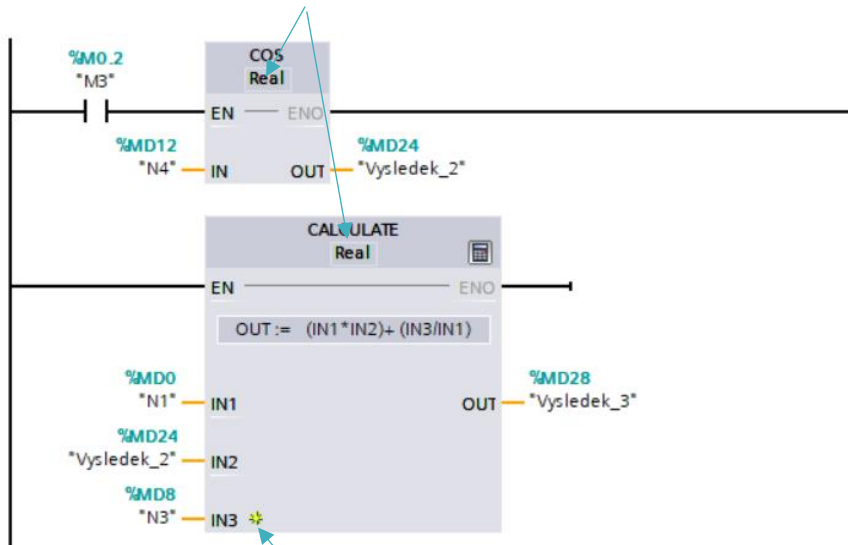
**Calculate** (spočítat)



S tímto blokem můžeme spočítat algebraické výrazy, které zadáme v podobě vzorce v okně bloku.

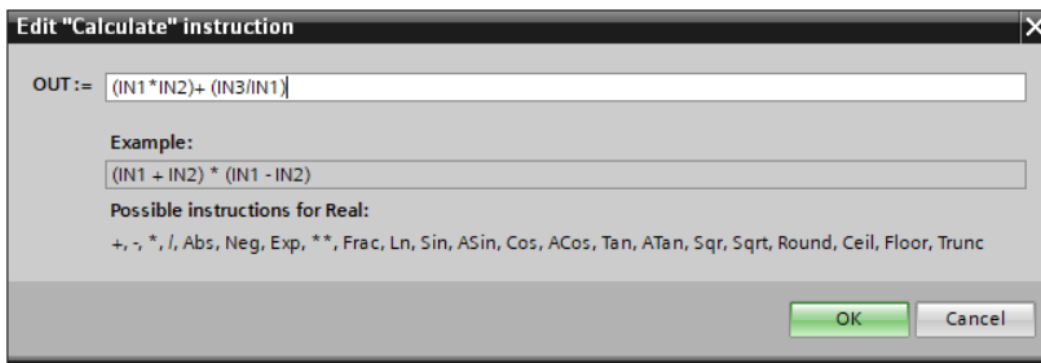
V následujícím bloku je ukázka zapojení tří vstupů (tří proměnných) do bloku „CALCULATE“ a jejich výpočet. Blok „COS“ je použitý z důvodu vstupu proměnné z jiného bloku. Hodnota COS se zadává v radiánech (na obrazovce byla zvolena hodnota  $1.047 \text{ rad} = 60^\circ$ ,  $\cos 60^\circ = 0,5$ ).

V poli pod názvem bloku vybereme datový typ, zde je to Real (desetinné číslo).



Počet vstupů rozšíříme kliknutím na hvězdičku

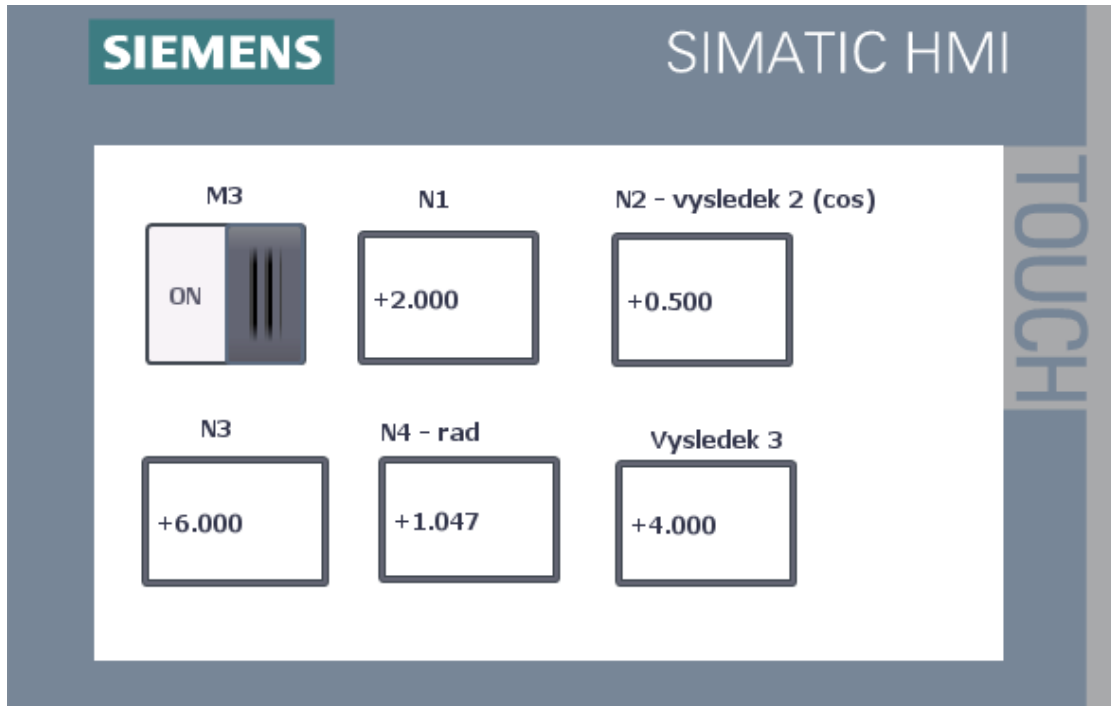
Ve druhém okně zadáme vzorec:



Zadání a výsledek je pro ilustraci zobrazen na HMI panelu.

V bloku CALCULATE jsme zadali:  $OUT = (NI1 * NI2) + (NI3/NI1)$ ,

$OUT = (2 \cdot 0,5) + (6 / 2) = 4$

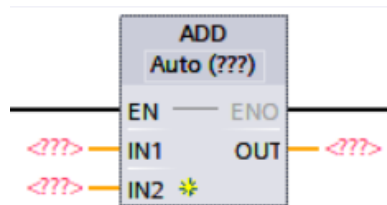


Adresy vstupů zadáme po kliknutí na otazníky. Otevře se tabulka „Default tag table“, kam jsme předem zadali názvy, datové typy a proměnné.

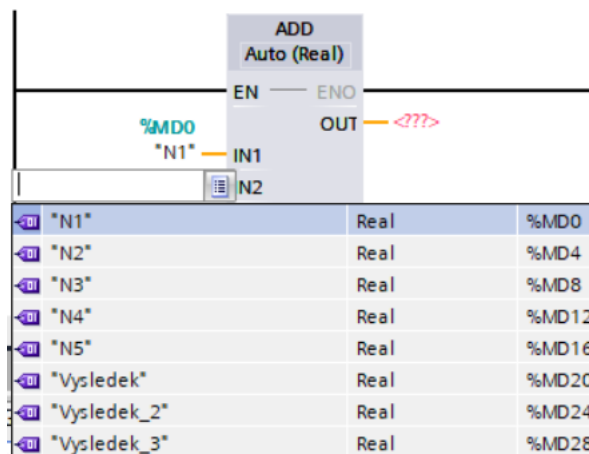
Tag Name	Data Type	Address
"N1"	Real	%MD0
"N2"	Real	%MD4
"N3"	Real	%MD8
"N4"	Real	%MD12
"N5"	Real	%MD16
"Vysledek"	Real	%MD20
"Vysledek_2"	Real	%MD24
"Vysledek_3"	Real	%MD28

## Add (addition) Sčítání

Sčítat můžeme sčítat více hodnot v závislosti na tom, kolik vstupů si nastavíme. Vstup přidáme kliknutím na hvězdičku u posledního vstupu. Zadávat můžeme i záporné hodnoty.

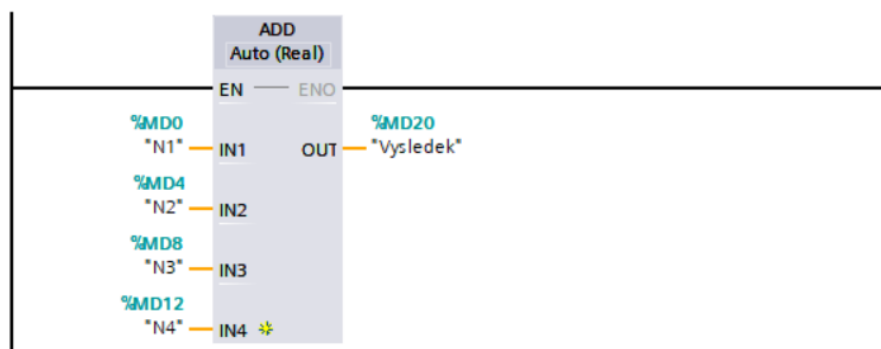


Vložení adres vstupů

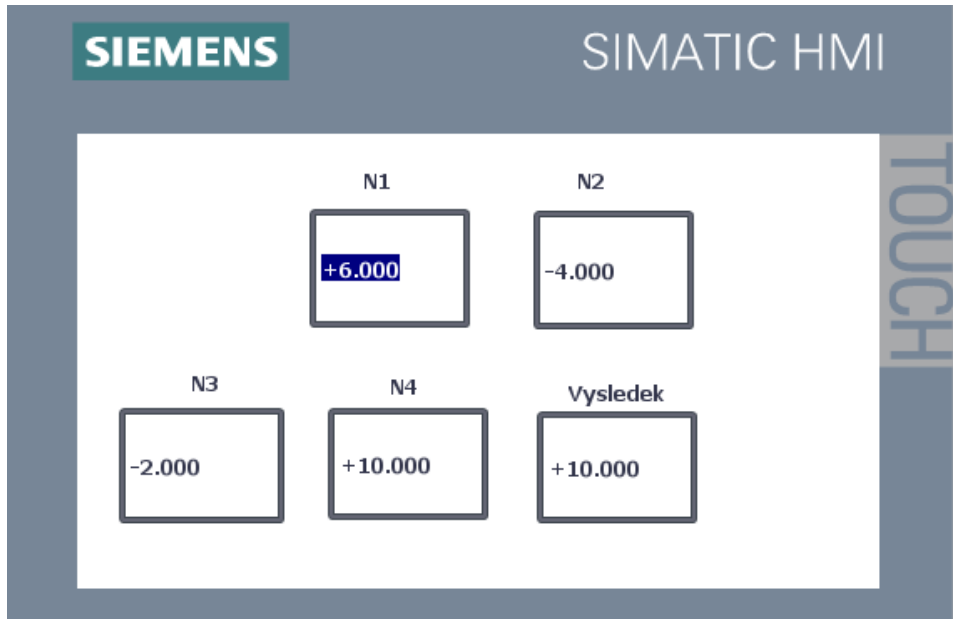


Výsledný blok ADD

$$OUT = IN_1 + IN_2 + IN_3 + IN_4$$

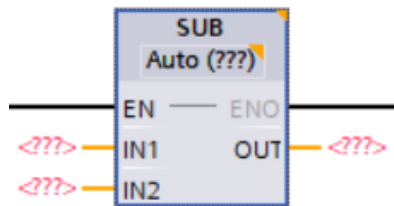


$$\text{OUT} = 6 - 4 - 2 + 10 = 10$$



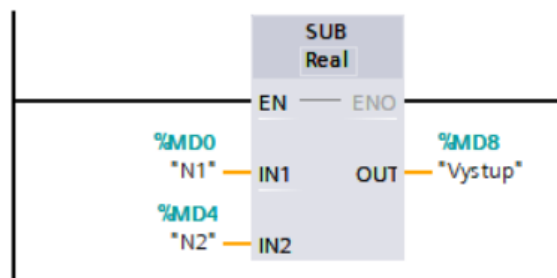
### SUB: Subtract (odečíst)

Blok SUB odečítá vstup IN2 od vstupu IN1.  $\text{OUT} = \text{IN1} - \text{IN2}$

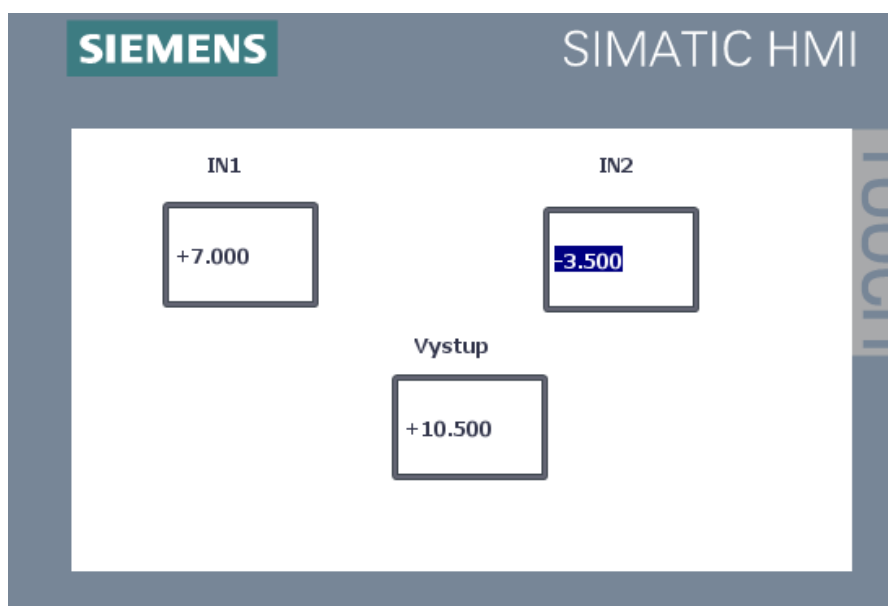


Kliknutím na otazníky se otevře okno pro zadání proměnných

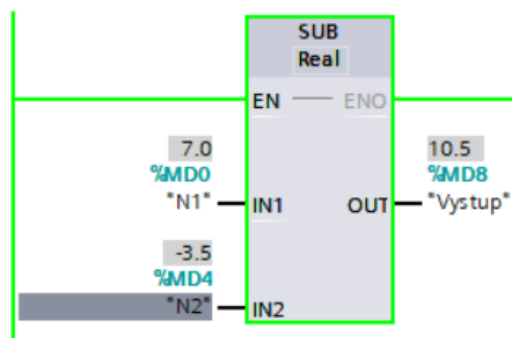
Symbolic Name	Symbolic Name	Symbolic Name	Symbolic Name
"Data_block_1"	Global DB	DB1	
"N1"	Real	%MD0	
"N2"	Real	%MD4	
"Vystup"	Real	%MD8	



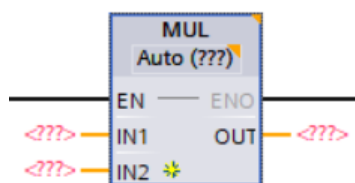
Zde je na panelu HMI zadaná hodnota IN1 = 7, IN2 = -3,5. Výsledek OU = 7 - (-3,5) = 10



Zadané hodnoty a výsledek můžeme vidět i na simulaci S 1200.

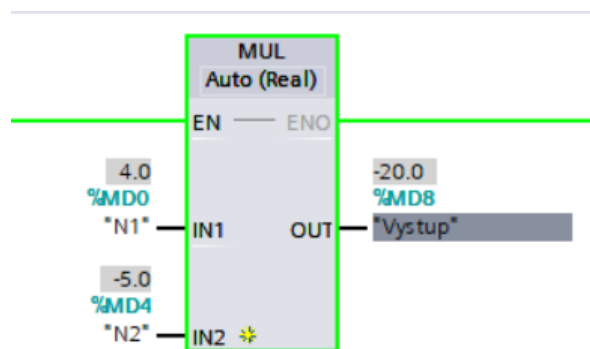
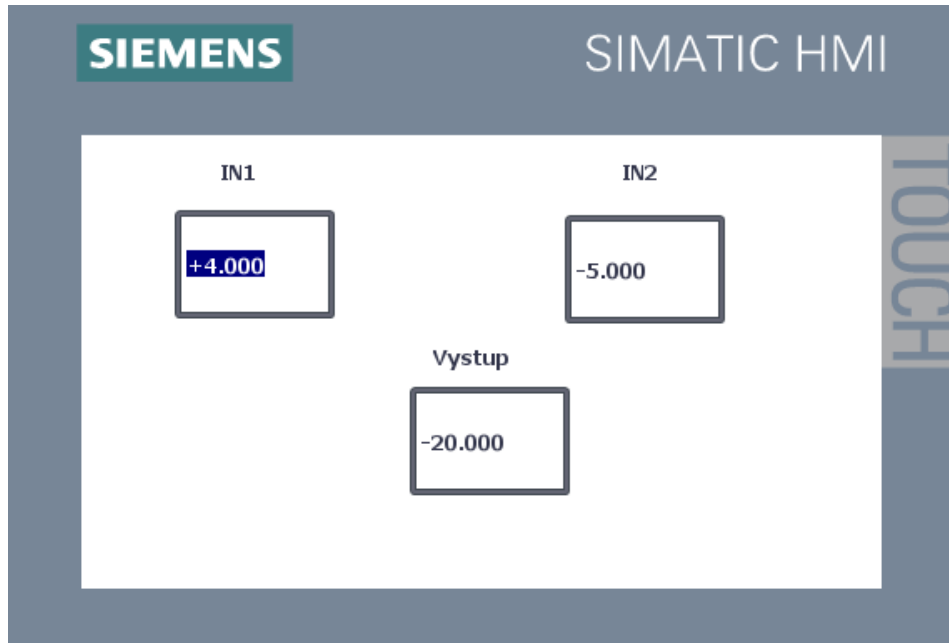


### MUL: Multiply (násobit)

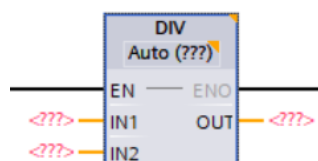


Blok násobí mezi sebou hodnoty vstupů IN1 a IN2 a výsledek zobrazuje na vstupu OUT.

Simulace příkladu  $OUT = 4 * (-5) = -20$

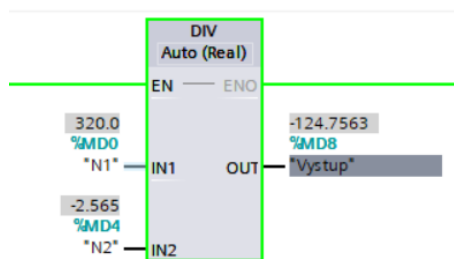
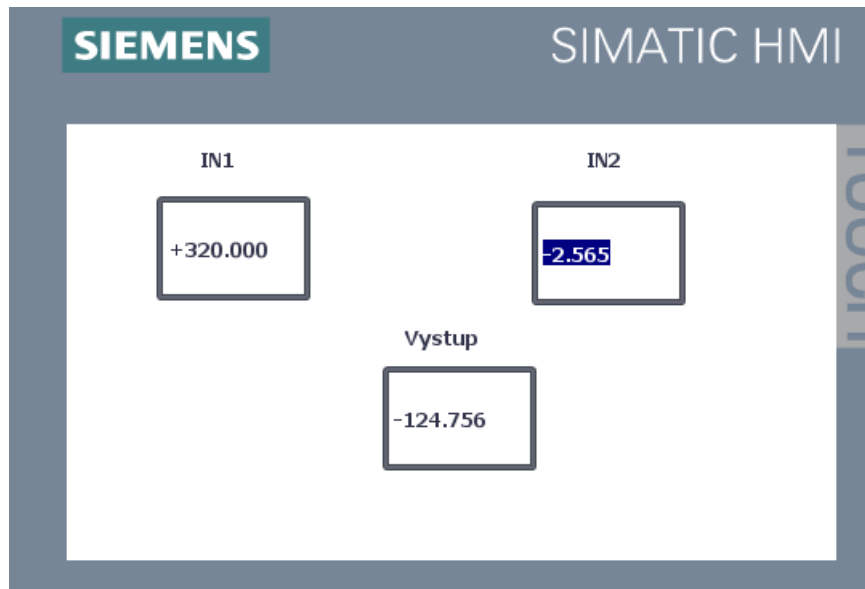


### DIV: Divide (dělení)



Dělení hodnoty vstupu IN1 vstupem IN2

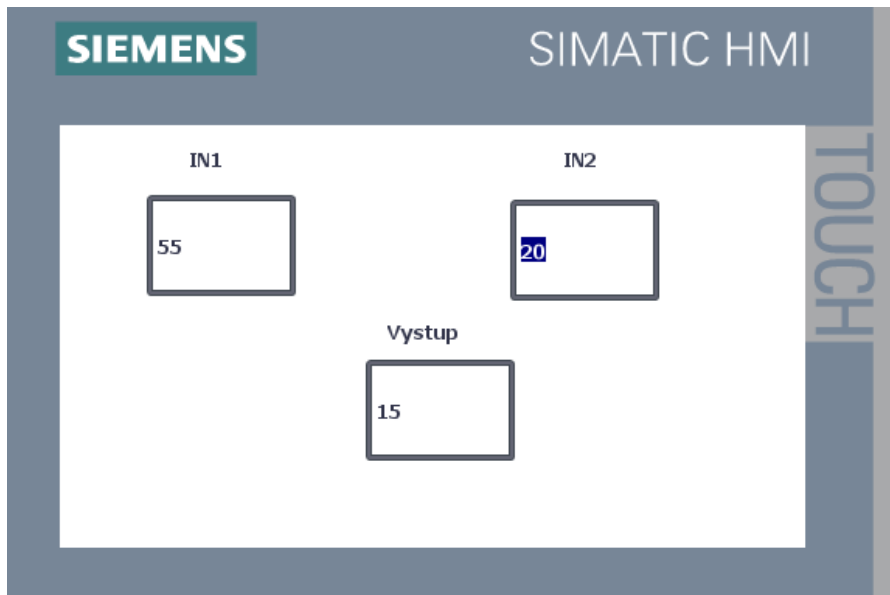
Příklad dělení čísla 320 : (-2,565) = -124,756



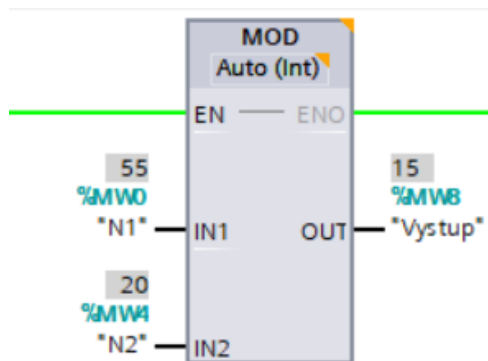
**MOD:** Return remainder of division ( vrácení zbytku po dělení)

Tato funkce pracuje s celočíselným vstupem Int. Na uvedeném příkladu je vstup IN1 55 a IN2 20.  $55 / 20 = (2 \times 20) + 15$ , číslo 15 je vrácený zbytek po dělení, který se zobrazí na výstupu OUT.

Zobrazení na panelu HMI

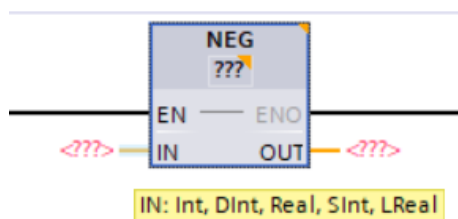


Simulace S 1200

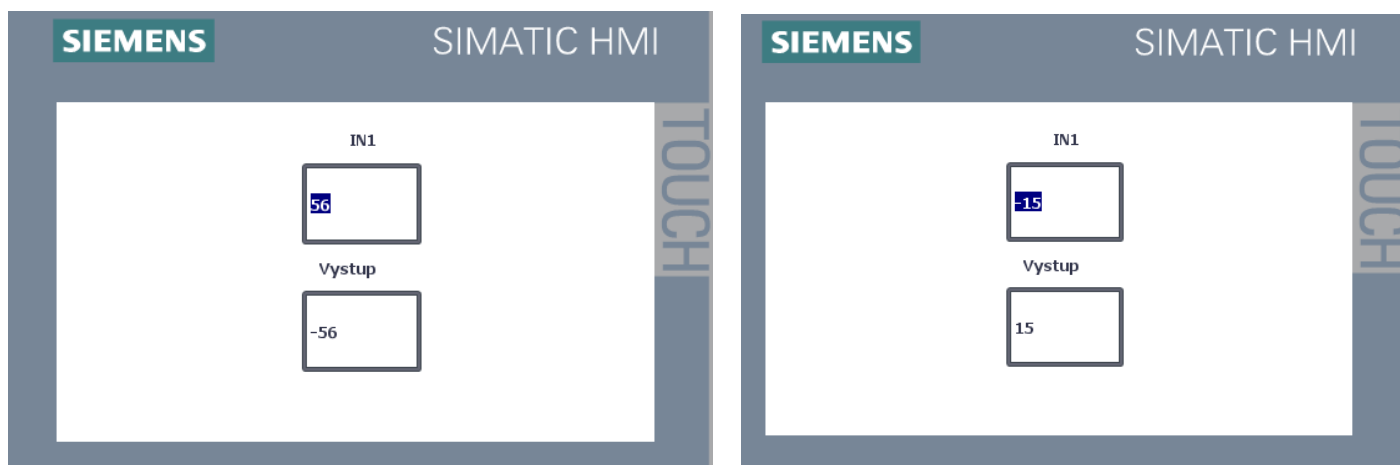


**NEG:** Create twos complement (vytvořit dvojkový doplněk)

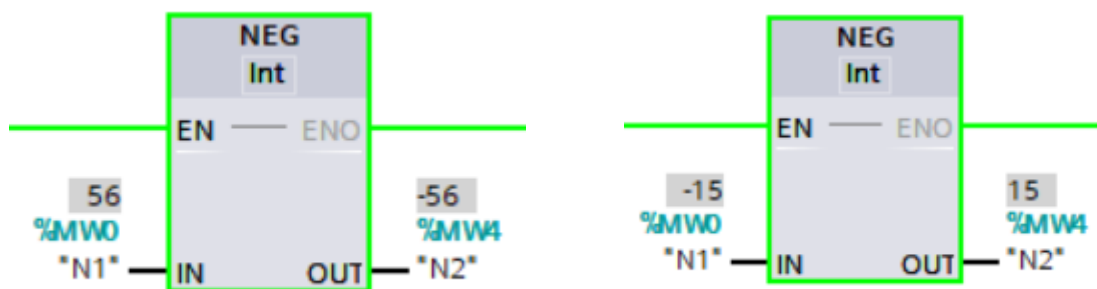
Blok NEG mění znaménka čísel na vstupu na čísla a výstupu . Je-li na vstupu kladné číslo např. +5 na výstupu bude záporné číslo -5. Najedeme-li šipkou myši na začátek vstupu nebo výstupu u jakéhokoliv bloku zobrazí se datové typy, které bude moc použít.



Zobrazení na panelu HMI



Simulace S1200

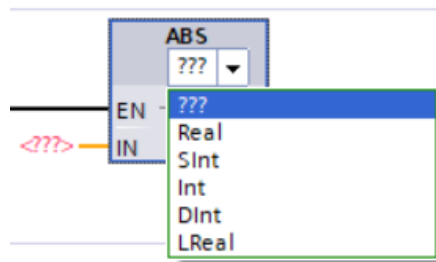


### ABS – absolutní hodnota

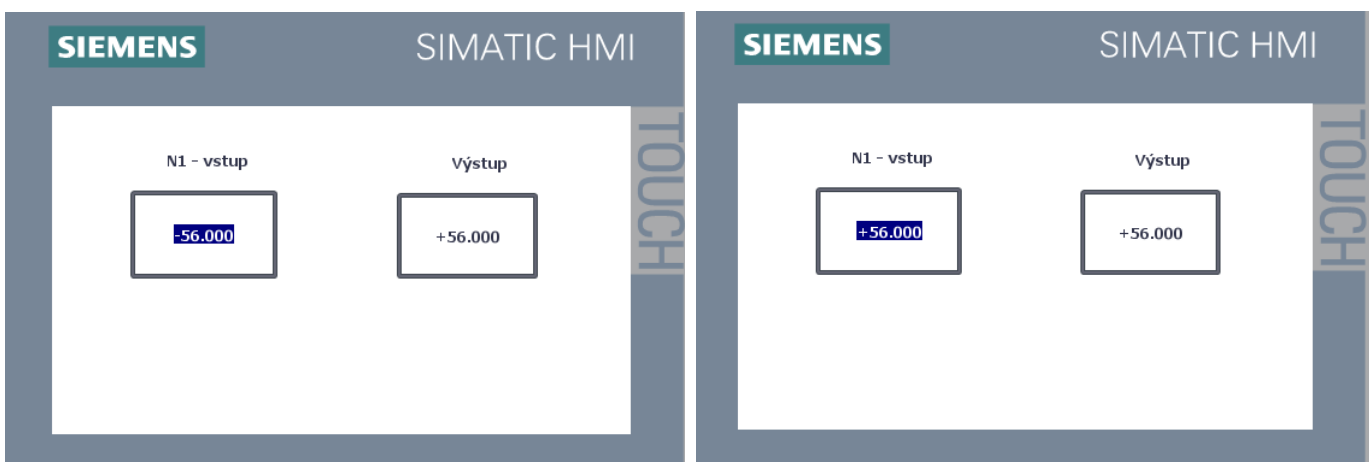
Absolutní hodnota čísla kladného zadaného na vstupu IN je na výstupu OUT to samé číslo, též kladné. Číslo záporné na vstupu je to samé číslo na výstupu kladné.

Absolutní hodnota z čísla je vždy číslo nezáporné, tedy větší nebo rovno nule. Pokud máme vypočítat absolutní hodnotu z čísla kladného, bude to vždy to samé číslo. Budeme-li ovšem chtít zjistit absolutní hodnotu ze záporného čísla, bude to číslo opačné.

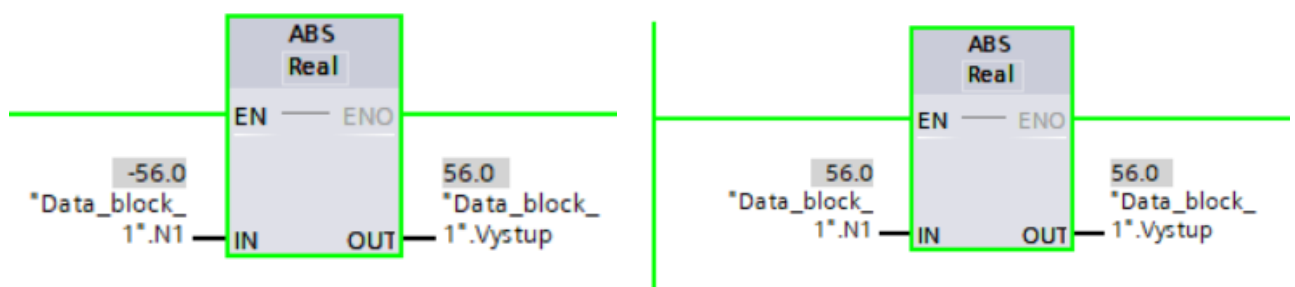
Kliknutím na otazníky pod nápisem ABS vybereme datový typ shodný s dat. typem na vstupu IN.



Simulace na pnelu HMI



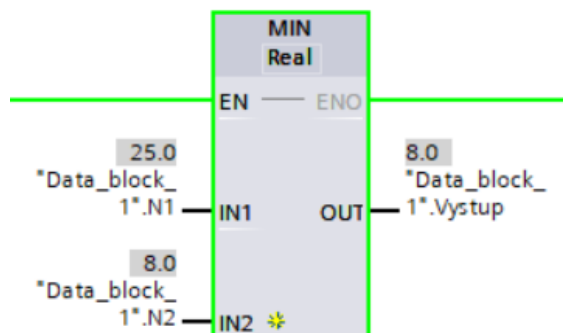
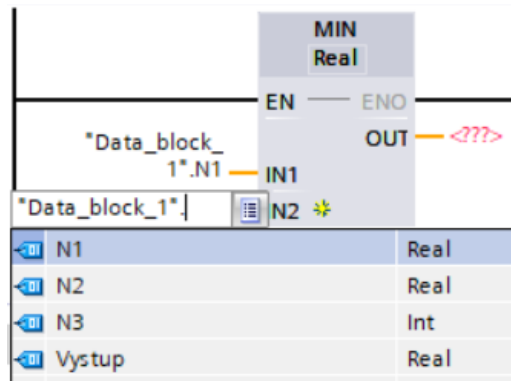
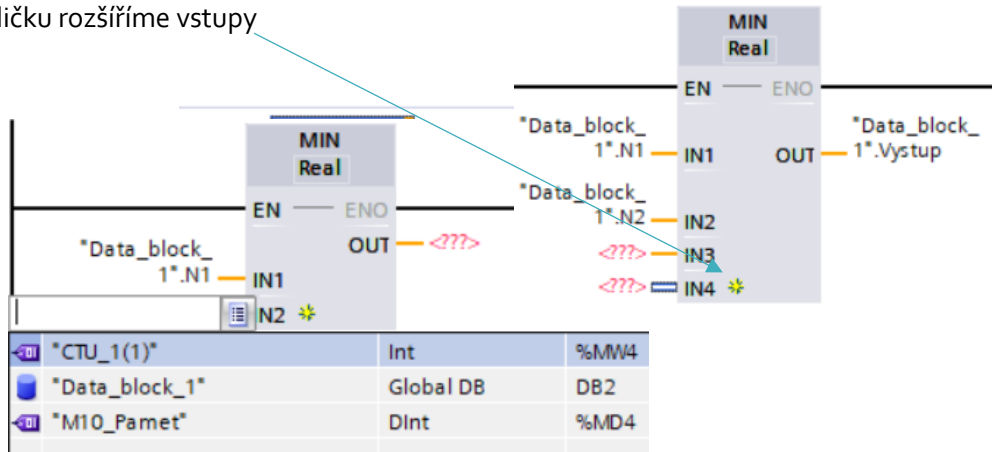
Simulace S1200

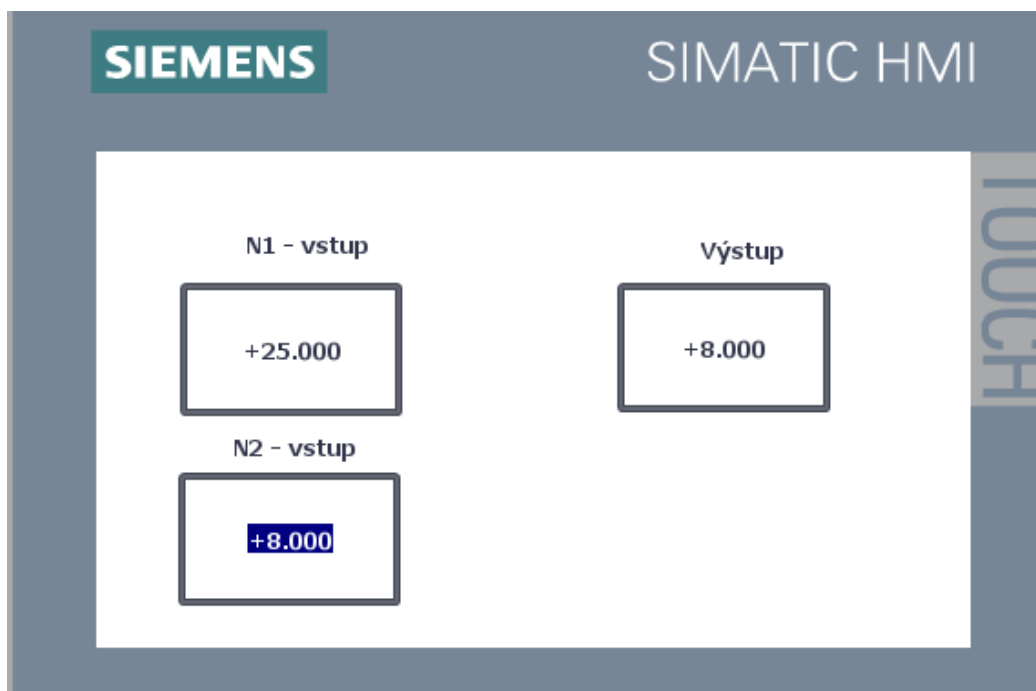


## MIN Get minimum

Instrukce "Get minimum" porovnává hodnoty na dostupných vstupech a zapisuje nejnižší hodnotu na výstup OUT. Počet vstupů lze na instrukčním boxu rozšířit o další vstupy. Vstupy jsou v rámečku očíslovány vzestupně.

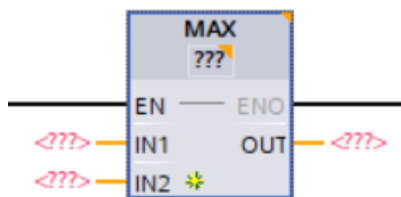
Kliknutím na hvězdičku rozšíříme vstupy



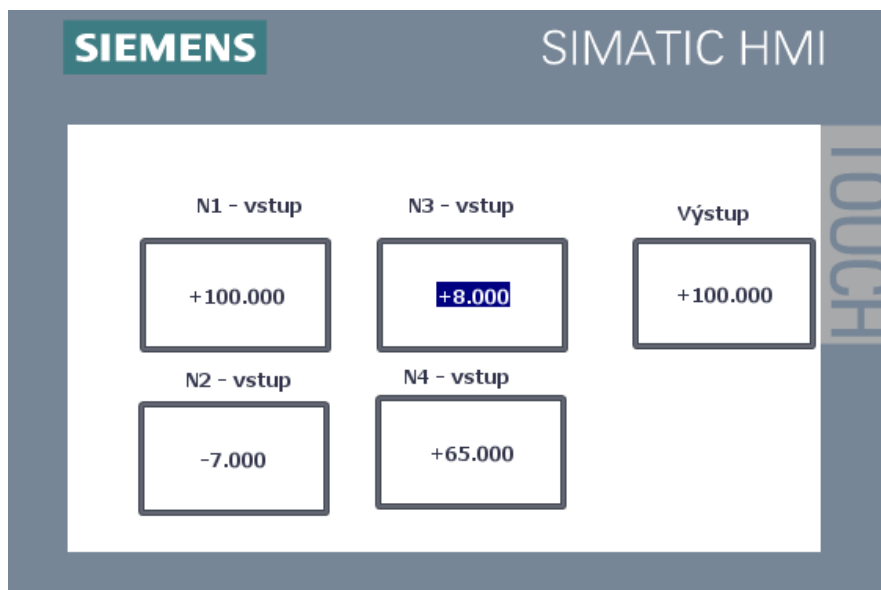


### Max Get maximum

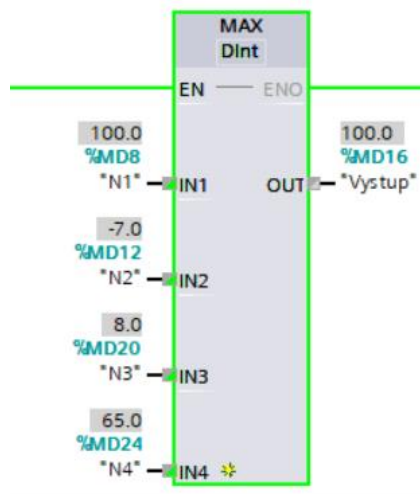
Instrukce "Get maximum" porovnává hodnoty na dostupných vstupech a zapisuje nejvyšší hodnotu na výstup OUT. Počet vstupů lze na instrukčním boxu rozšířit o další vstupy. Vstupy jsou v rámečku očíslovány vzestupně. Pro provedení instrukce musí být specifikovány minimálně dva a maximálně 100 vstupů.



Na panelu HMI jsou čtyři vstupy s hodnotami 100, 8, -7, 65, na výstup se zapíše největší hodnota a to je 100.



Simulace S1200

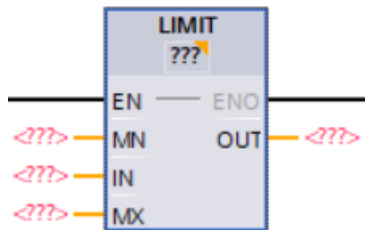


**LIMIT:** Set limit value (Limitní hodnota)

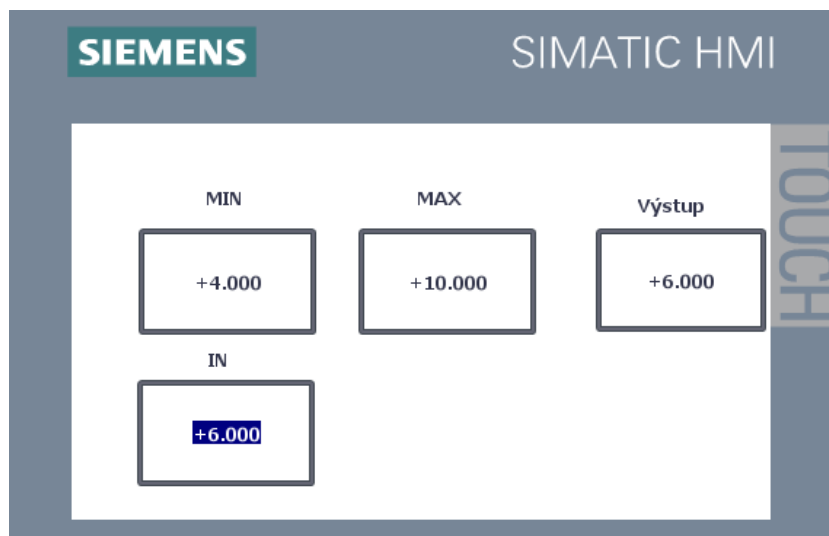
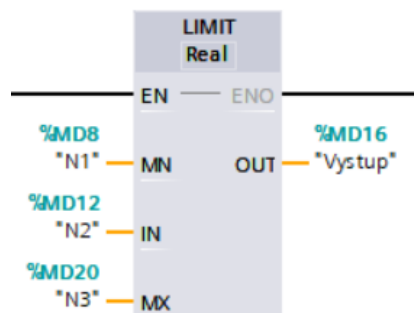
Pomocí instrukce "Nastavit limitní hodnotu" můžete omezit hodnotu na vstupu IN na hodnoty na vstupech MN a MX. Pokud hodnota na vstupu IN splňuje podmínku  $MN \leq IN \leq MX$ , je zkopírována na výstup OUT. Pokud není podmínka splněna a vstupní hodnota IN je pod dolní mezí MN, je výstup OUT nastaven na hodnotu vstupu MN. Při překročení horní hranice MX se výstup OUT nastaví na hodnotu vstupu MX.

$MIN \leq IN \leq MAX$ , potom  $OUT = IN$   
 $IN \leq MIN$ , potom  $OUT = MIN$   
 $IN \geq MAX$ , potom  $OUT = MAX$

MIN	MAX	IN	OUT
4	10	5	5
4	10	2	4
4	10	12	10

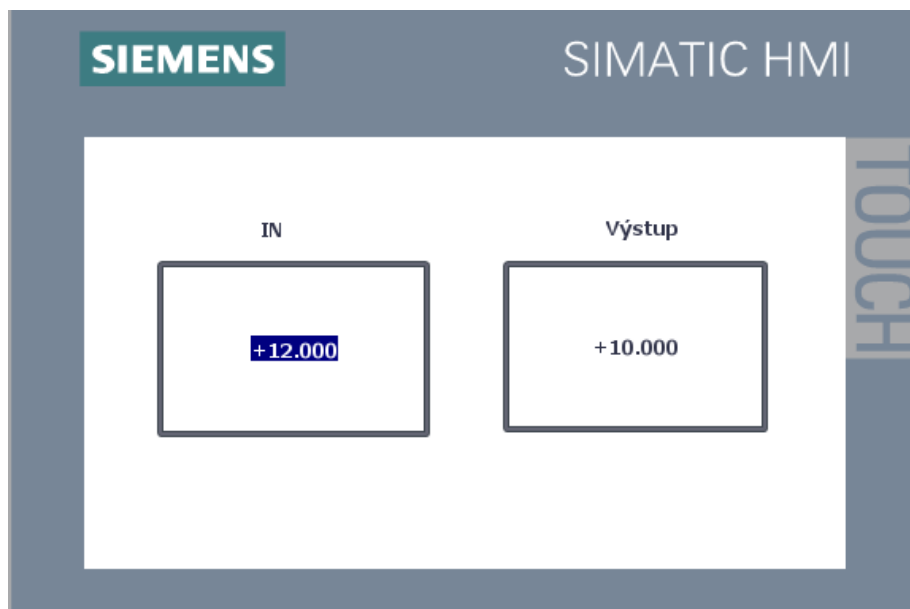


Zapojení vstupů bloku, kde jsou všechny proměnné. Na panelu HMI můžeme měnit jejich hodnoty. Vstup může být např. čítač. Zde je použitý panel HMI z důvodu jednoduchého popisu bloku.



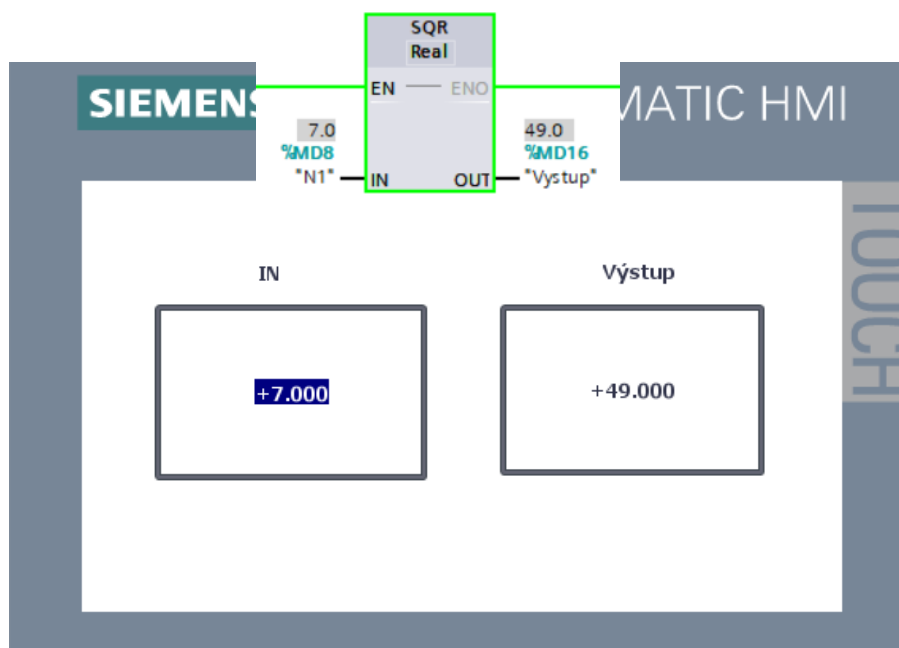
Zde jsou vstupy MN a MX zadané jako konstanty, které je možné měnit pouze v programu.

Simulace programu s konstantami MN a MX zobrazená na panelu HMI.



### SQR: Form square (druhá mocnina)

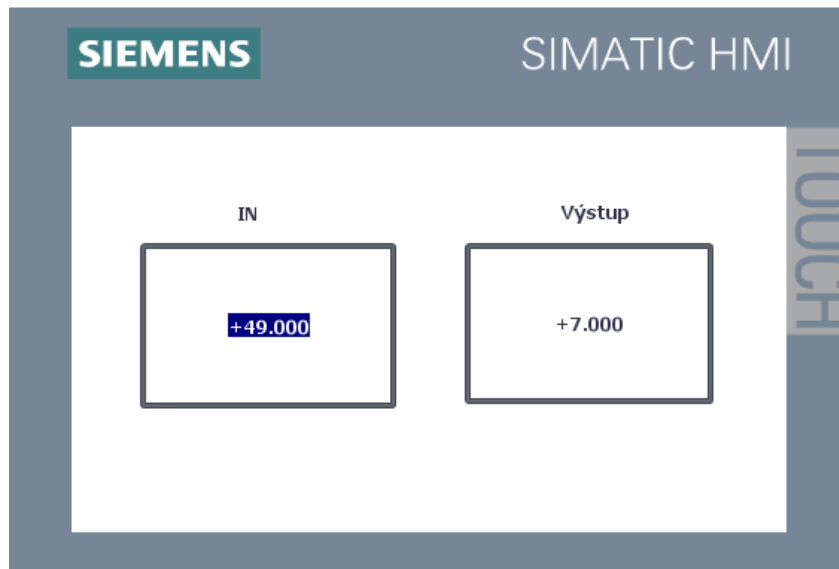
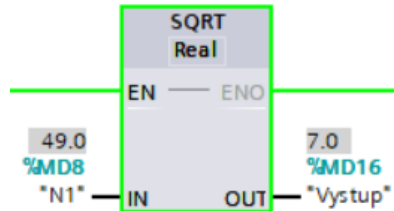
Vstupní hodnota IN se umocní na druhou ( $x^2$ ) a tato hodnota je výstup. Na příkladu je  $7^2 = 49$



## SQRT: Form square root (druhá odmocnina)

$$\sqrt{x}$$

Na příkladu je druhá odmocnina ze 49.



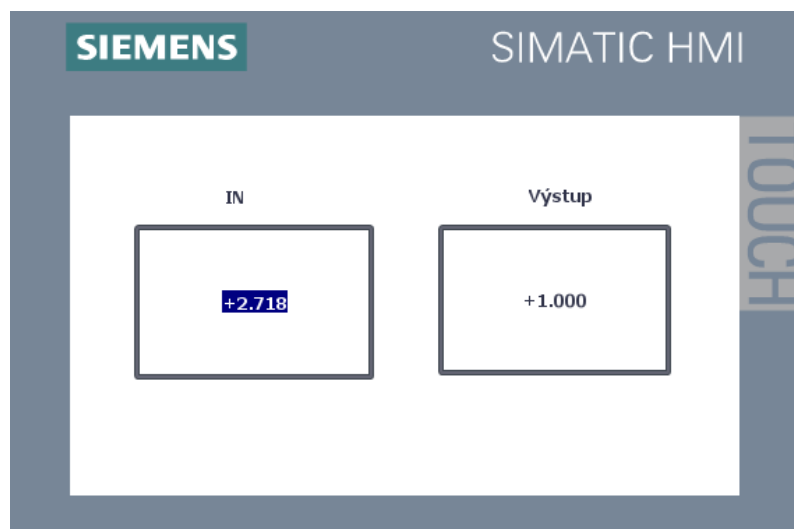
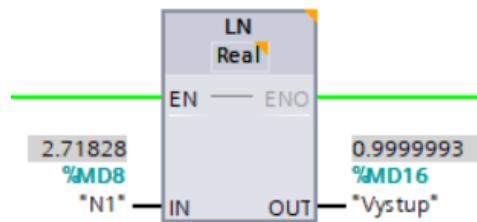
## LN: Form natural logarithm (přirozený logaritmus)

Pro výpočet přirozeného logaritmu se základem  $e$  ( $e = 2,718282$ ) hodnoty na vstupu IN můžete použít instrukci "Form natural logaritmu". Výsledek je odeslán na výstup OUT a lze jej zde dotazovat. Instrukce má kladný výsledek, pokud je vstupní hodnota větší než nula. Pokud jsou vstupní hodnoty menší než nula, výstup OUT vrátí neplatné číslo s plovoucí desetinnou čárkou.

$$y = \ln x \Leftrightarrow e^y = x$$

$$\text{OUT} = \ln \text{IN}$$

V příkladu je zvolen výpočet přirozeného logaritmu o hodnotě jeho základu, což je (e = 2,718282). Protože logaritmus čísla IN je exponent OUT, kterým musíme základ umocnit, abychom dostali logaritmované číslo, vyjde nám hodnota 1.



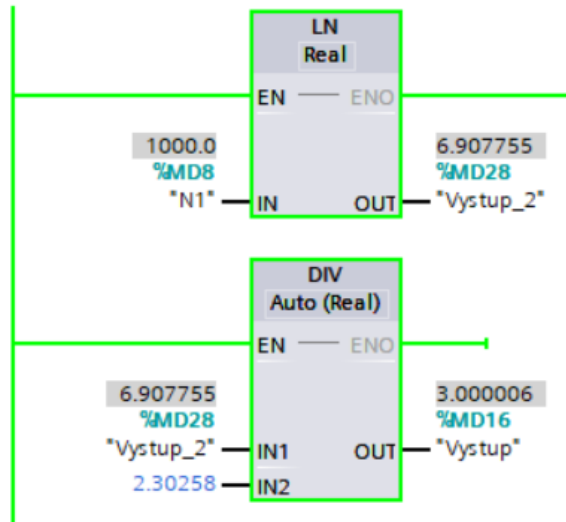
### Výpočet dekadického logaritmu log o základu 10

Blok pro dekadický logaritmus v nabídce není, proto si hodnotu dekadického logaritmu vypočteme pomocí přirozených logaritmů.

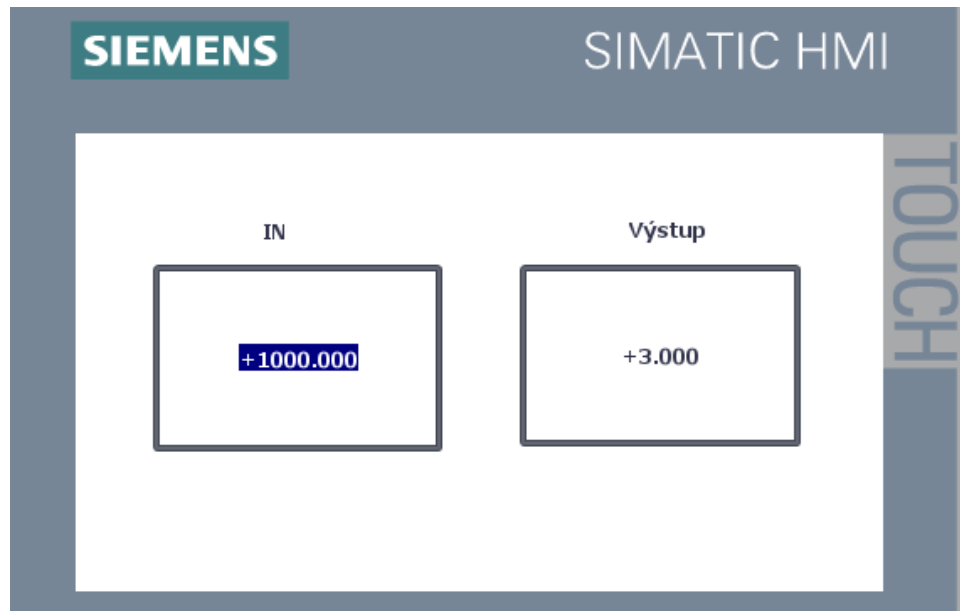
$$\log_a x = \frac{\ln x}{\ln a}$$

Př.:  $\log 1000$        $\log 1000 = \frac{\ln 1000}{\ln 10} = \frac{6,9}{2,3} = 3$

K výpočtu použijeme blok LN pro přirozený logaritmus a blok DIV dělení. V bloku LN se vypočte  $\ln 1000$  a v bloku DIV se provede podíl výstupu z bloku LN s konstantní hodnotou pro LN 10 tj. 2,3.



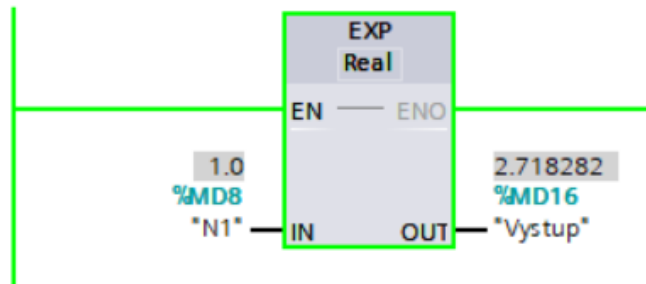
Simulace na panelu HMI



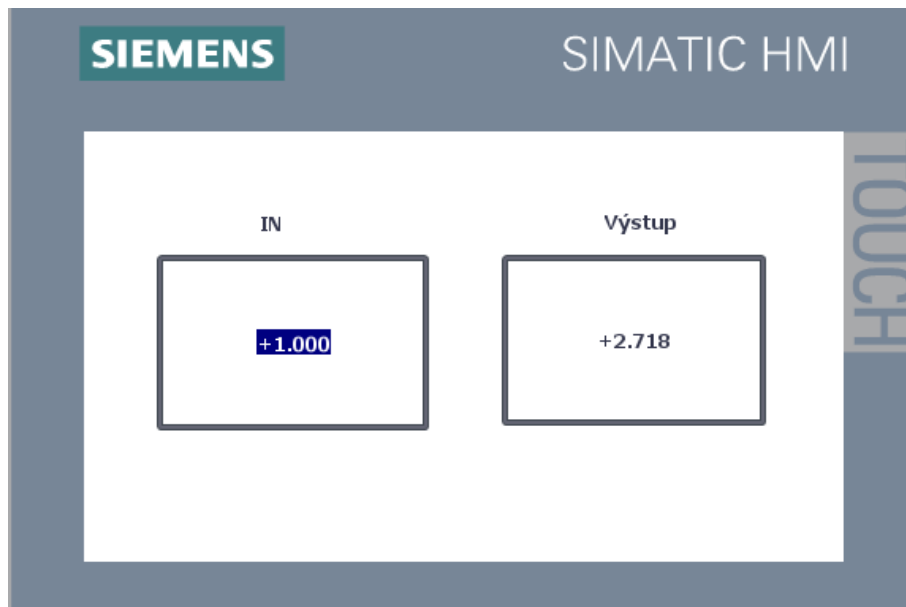
**EXP:** Form exponential value - umocněný základ e ( $e = 2,718282$ ) na x

$$y = e^x$$

V příkladu je zvolené  $x = 1$  ( $e^1 = e = 2,718$ ), pro snadné ověření fungování výpočtu.



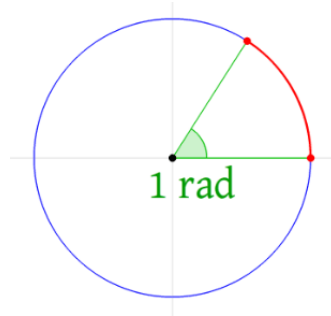
Simulace na panelu HMI



## Goniometrické funkce

Vstupní proměnná IN goniometrických funkcí se do programových bloků S1200 zadává v radiánech

$\pi$  radián je středový úhel, který přísluší oblouku o stejné délce, jako je poloměr kružnice. Je to jednotkový úhel při měření v **obloukové míře**.



Obvod kružnice je  $2\pi r$ , na obvod kružnice se vejde  $2\pi$  radiánů.  $2\pi$  radiánů náleží úhlu  $360^\circ$ . Jeden radián odpovídá hodnotě  $360^\circ/2\pi \approx 57,296^\circ \approx 57^\circ 17' 45''$ . Jeden stupeň je  $2\pi/360 \approx 0,01745$  rad.

Přepočet stupňů na radiány:  $a = \alpha \cdot \pi / 180$ , př.  $\alpha = 60^\circ$ ,  $a = 60 \cdot \pi / 180 = 0,52$  rad

Přepočet radiánů na stupně:  $\alpha = a \cdot 180 / \pi$

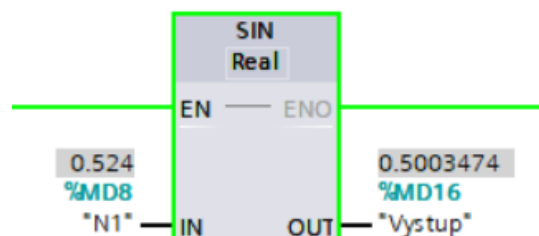
Tabulka

úhel ve °	0	5	10	15	20	30	35	40	45	50	55	60	65	70	75	80	85	90
úhel v rad	0,000	0,087	0,175	0,262	0,349	0,524	0,611	0,698	0,785	0,873	0,960	1,047	1,134	1,222	1,309	1,396	1,484	1,571
sin	0,000	0,087	0,174	0,259	0,342	0,500	0,574	0,643	0,707	0,766	0,819	0,866	0,906	0,940	0,966	0,985	0,996	1,000
cos	1,000	0,996	0,985	0,966	0,940	0,866	0,819	0,766	0,707	0,643	0,574	0,500	0,423	0,342	0,259	0,174	0,087	0,000
tg	0,000	0,087	0,176	0,268	0,364	0,577	0,700	0,839	1,000	1,192	1,428	1,732	2,144	2,747	3,732	5,671	11,427	-

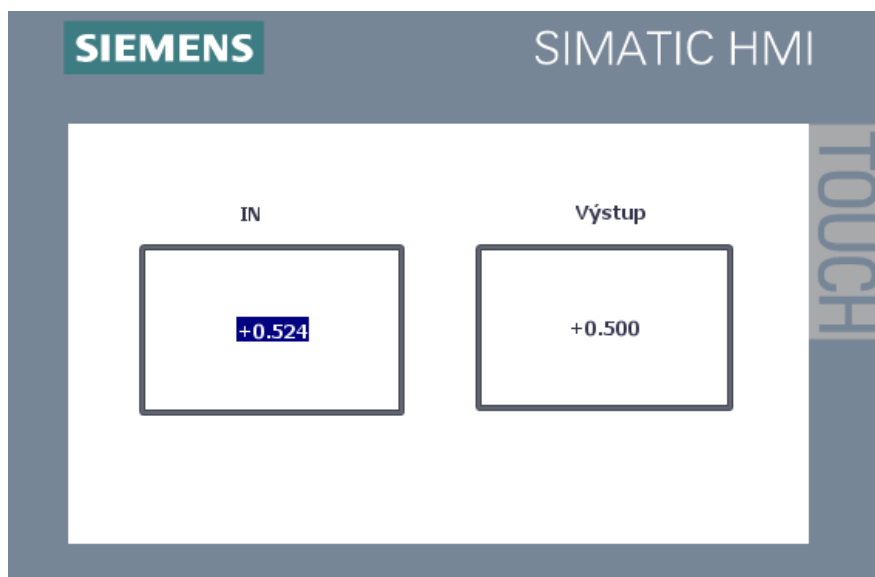
## SIN: Form sine value (funkce sinus)

Zadávání v radiánech

Na vstupu IN je zadána hodnota 0,524 rad. Výstup OUT  $\sin 0,524 = 0,5$

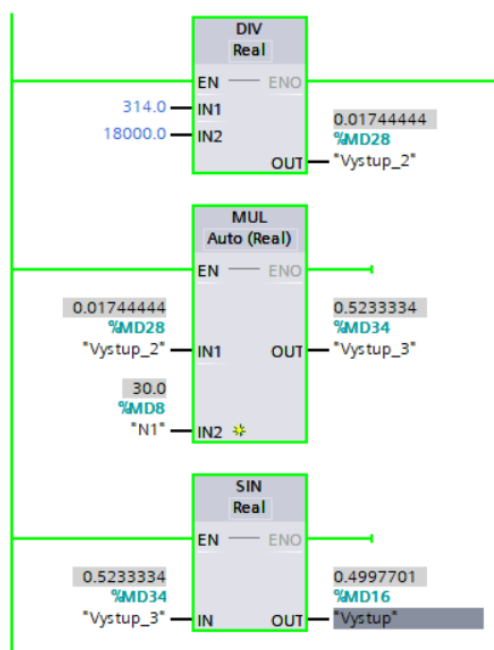


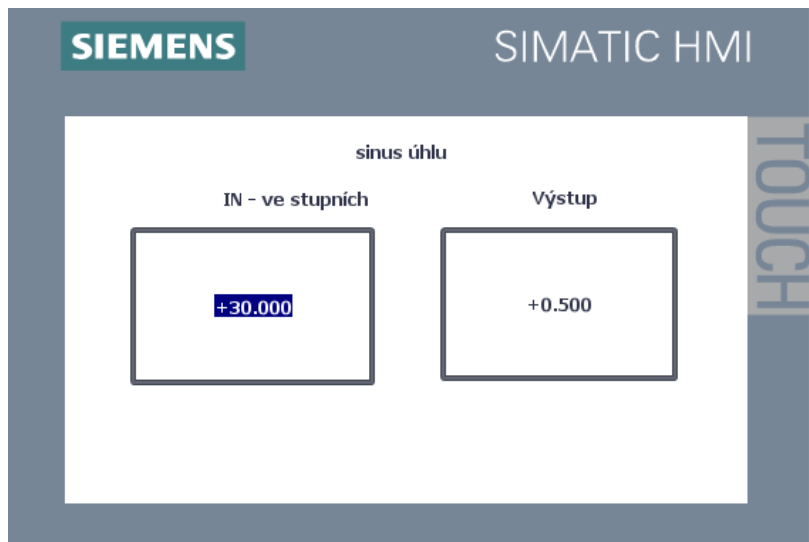
Zobrazení na panelu HMI. V okně IN se zadává hodnota v radiánech.



#### Zadávání ve stupních

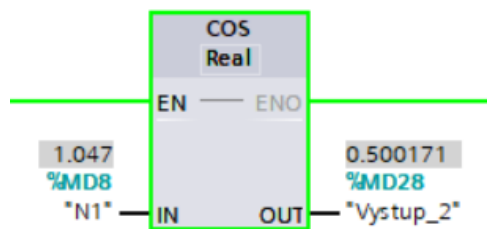
Protože není možné zadávat stupně přímo, musí se provést přepočítání stupňů na radiány. Jak již bylo v úvodu popsáno  $\text{rad} = \alpha \cdot \pi / 180$ . Na vstupu do bloku nelze zadat desetinné číslo, proto vstupy bloku **DIV** (dělení) zadáme jako jejich násobek 100, abychom dostali celá čísla. Výstup z bloku DIV bude vstupem do bloku **MUL** (násobení). Zde se vynásobí výstup z DIV ( $314/18000$ ) se vstupem zadaným ve stupních ( $\alpha \cdot 314/18000$ ). Výstup z bloku DIV bude v radiánech. Vstup do bloku SIN je výstup z bloku MUL, tedy radiány.



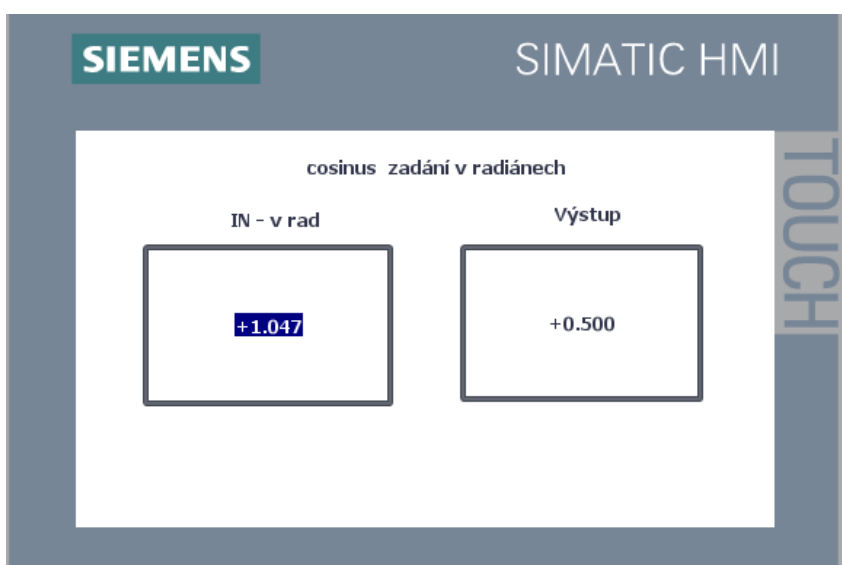


### COS: Form cosine value - funkce cosinus

Na vstupu IN je zadána hodnota v rad (1.047) výsledek je 0,5.

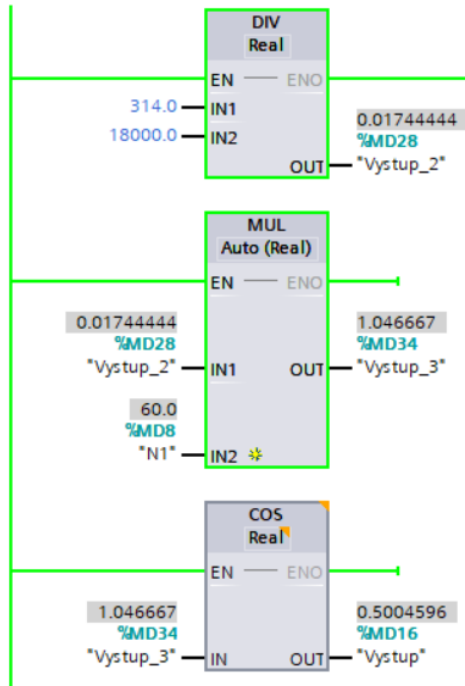


Výsledek a zadání na panelu HMI

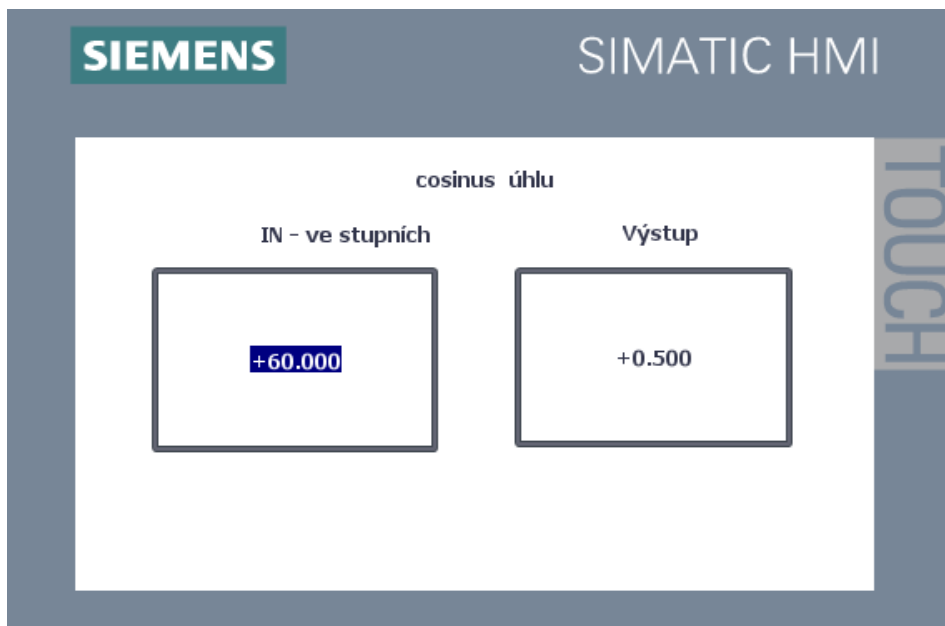


## Zadávání ve stupních

Jak bylo popsáno u funkce sinus, musí se stejným způsobem provést přepoččet stupňů na radiány.

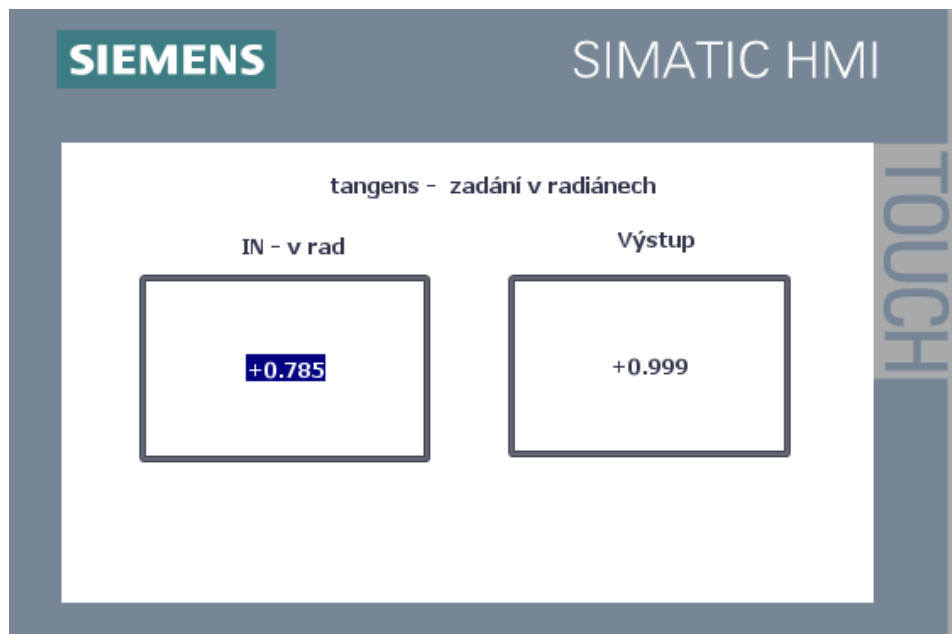
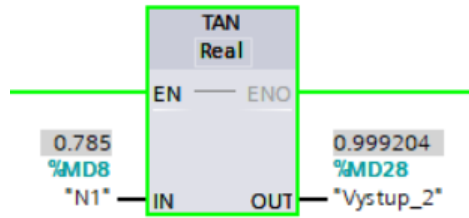


Zobrazení na panelu HMI pro úhel 60°

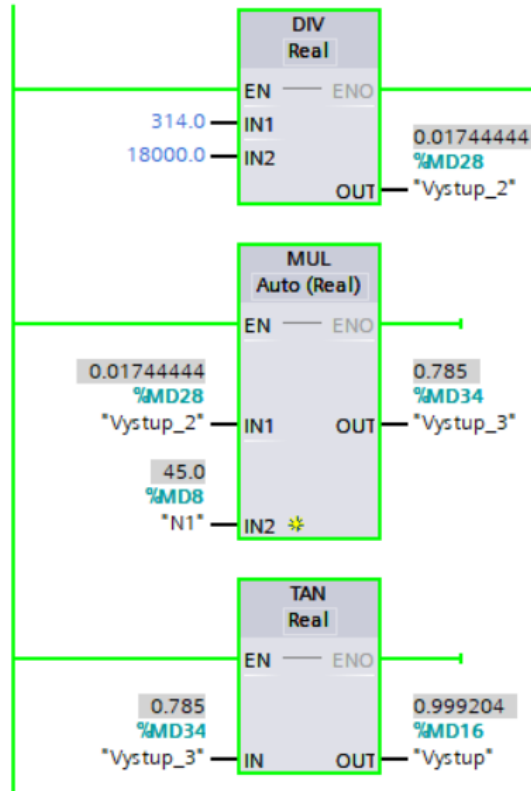


## TAN: Form tangent value - funkce tangens

Zadání v radiánech



## Zadání ve stupních



The image shows a SIMATIC HMI screen with the following layout:

- SIEMENS** logo in the top left corner.
- SIMATIC HMI** text in the top right corner.
- TOUCH** vertical label on the right side of the screen.
- tangens - zadání ve stupních** title centered at the top of the main display area.
- IN - stupně** label above a rectangular input field containing the value **+45.000**.
- Výstup** label above a rectangular output field containing the value **+0.999**.

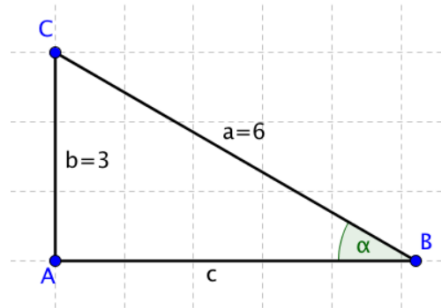
## Inverzní goniometrické funkce

Aby měly základní goniometrické funkce smysl, je potřeba definovat také jejich inverzní funkce. Ty se právě nazývají arcus- funkce. v případě bloků v programu pro S1200 se jedná o arcus sinus, arcus cosinus, arcus tangens. Obvykle zapisované jako arcsin, arccos, arctan. Je to hodnota poměru stran z níž určíme velikost úhlu.

$$\sin \alpha = b/a \Rightarrow \arcsin (b/a) = \alpha$$

$$\cos \alpha = c/a \Rightarrow \arccos (c/a) = \alpha$$

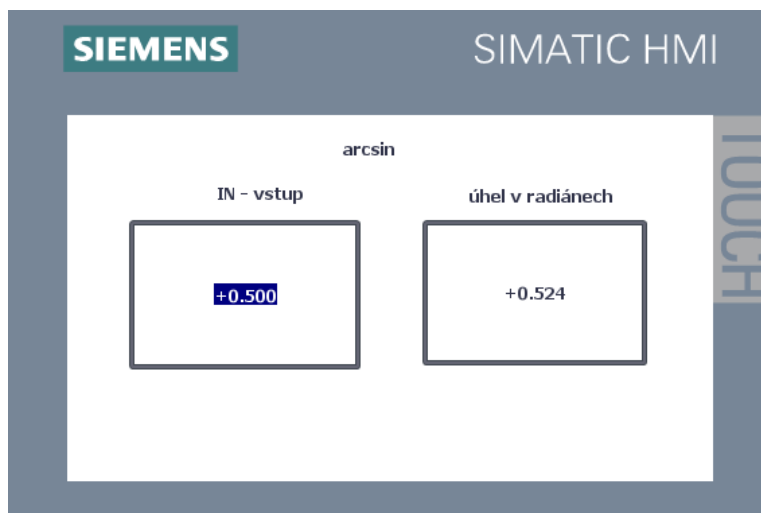
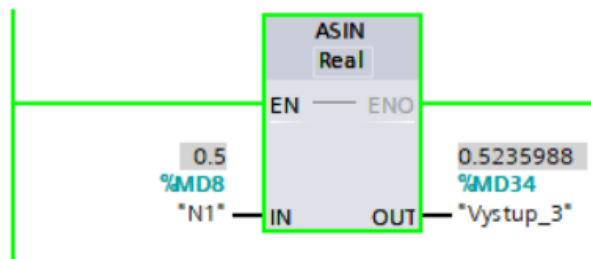
$$\operatorname{tg} \alpha = b/c \Rightarrow \operatorname{arctg} (b/c) = \alpha$$



### ASIN: Form arcsine value – funkce arcsin

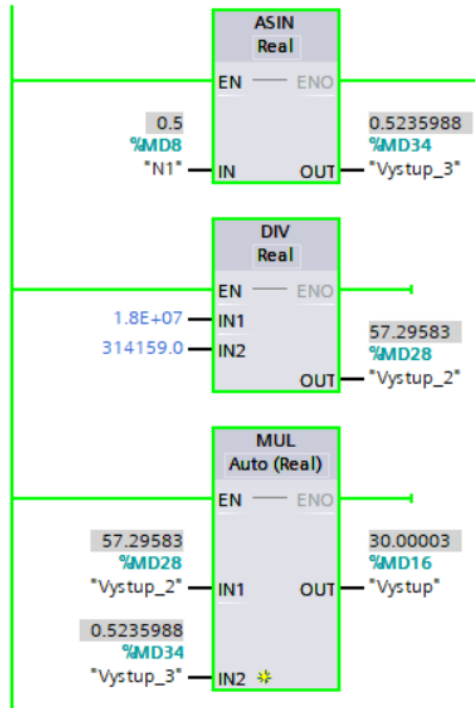
Výsledný úhel je v radiánech

Na obr. trojúhelníku je  $b/a = 3/6 = 0,5$ , tato hodnota je dosazena na vstup IN, výstup OUT je 0,5235 rad. Přepočet na stupně:  $\alpha = \text{OUT} \cdot 180 / \pi = 30^\circ$

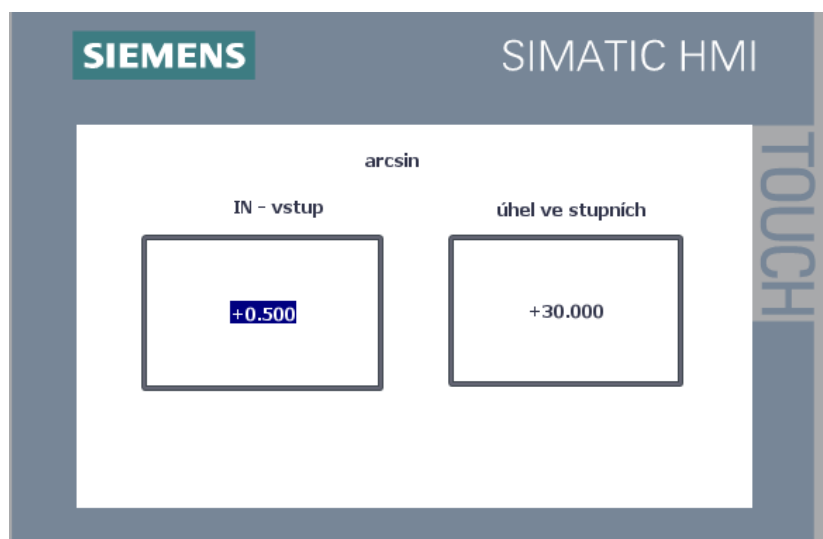


## Výpočet arcsin ve stupních

Nejprve se vypočte v bloku ASIN hodnota úhlu v radiánech. V bloku DIV (dělení) se vypočte  $180/\pi$  (čísla jsou pro větší přesnost výpočtu rozšířena na  $10^5$ ) a v bloku MUL je vynásobený výstup DIV a výstup z ASIN.

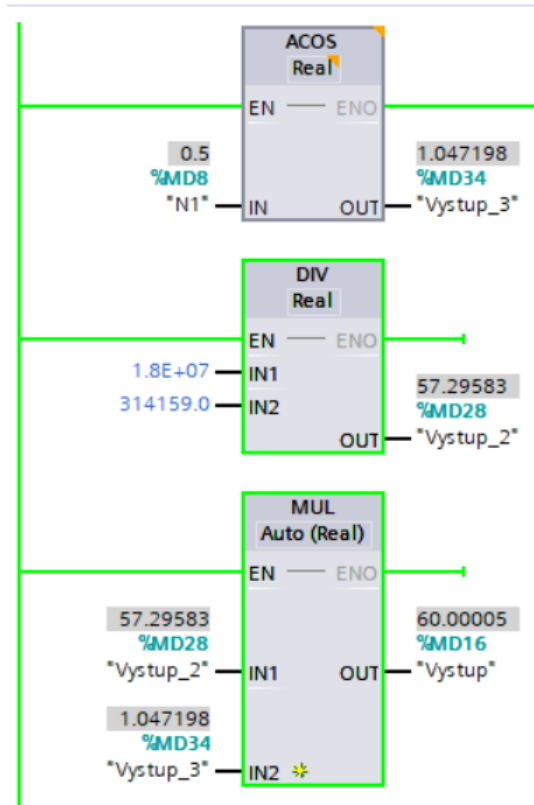


## Zobrazení na panelu HMI



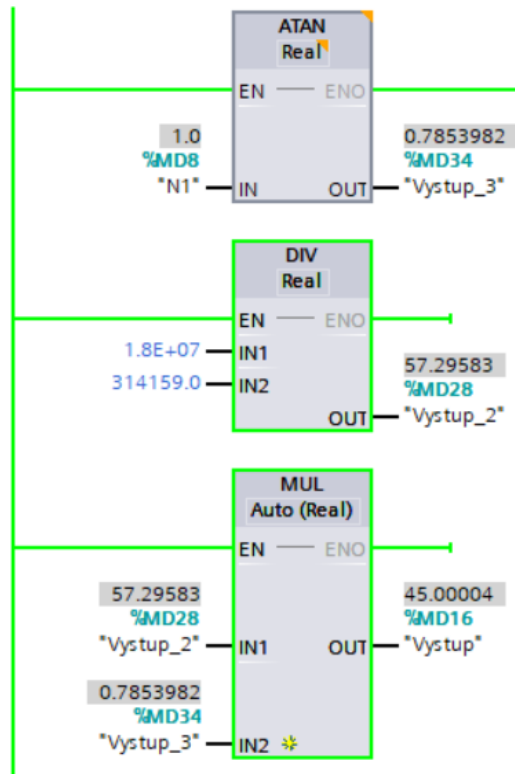
## ACOS: Form arccosine value – funkce arccos

Blok ACOS dává výstup v radiánech. Výstup z bloku MUL je ve stupních. Postup výpočtu převodu radiánů na stupně je shodný jako v předcházejícím příkladu pro ASIN. Na zapojení bloků je výpočet  $\arccos 0,5 = 60^\circ$



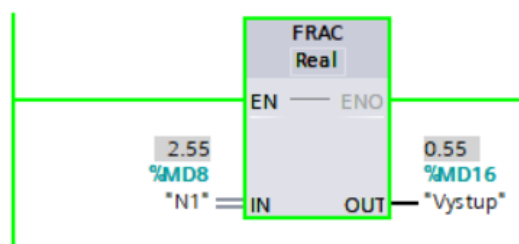
## ATAN: Form arctangent value – funkce arctg

Výstup bloku ATAN je v radiánech. Výstup z bloku MUL je ve stupních. Postup výpočtu převodu radiánů na stupně je shodný jako v příkladu pro ASIN. Na zapojení bloků je výpočet  $\arccos 1 = 45^\circ$



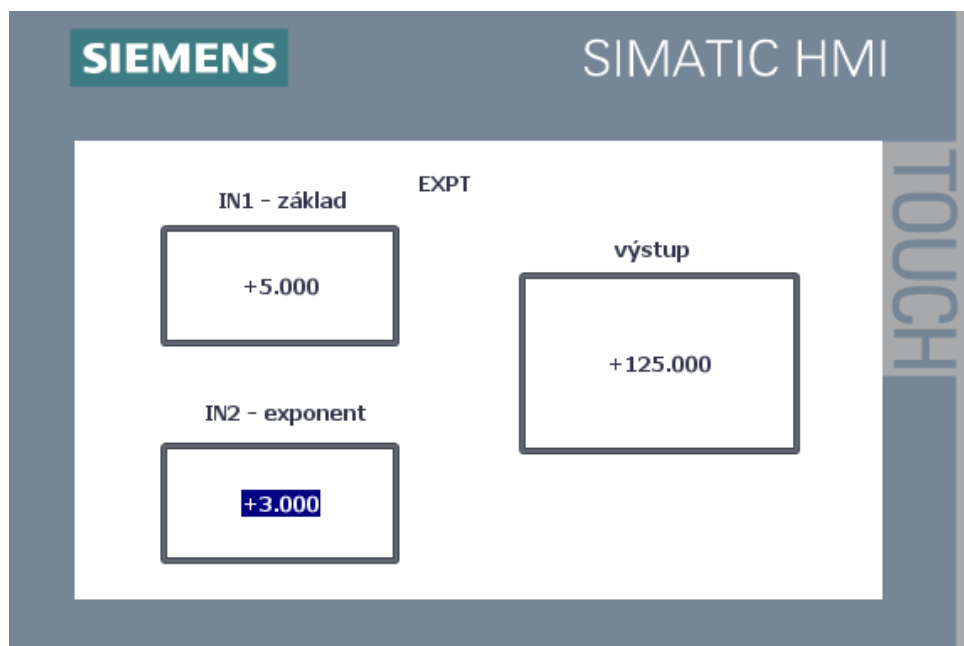
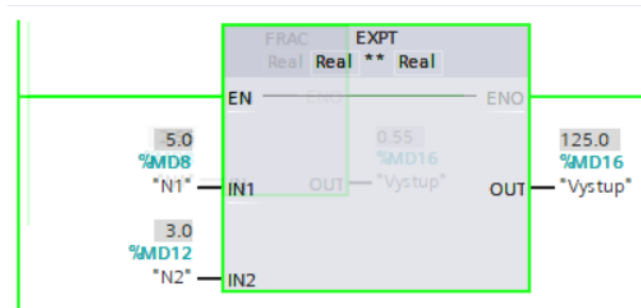
## FRAC: Return fraction – vrací číslo za desetinou čárkou

Př: Vstup IN = 2,55, výstup OUT = 0,55



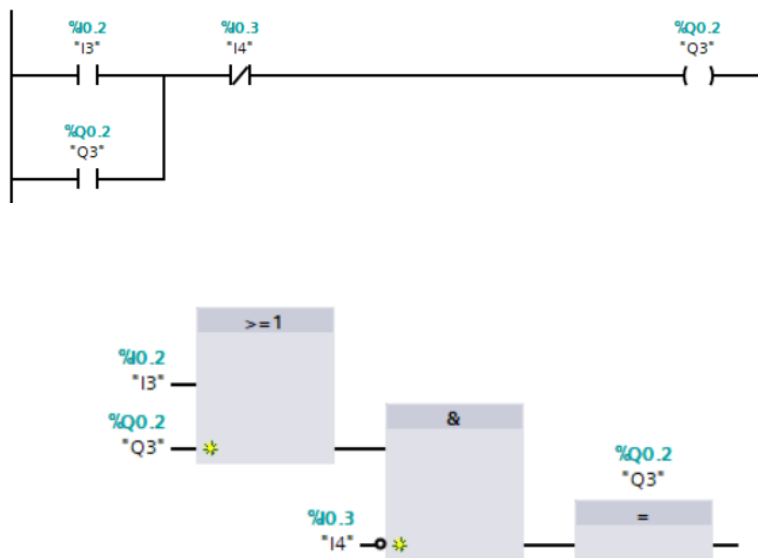
## EXPT: Exponentiate - umocnit

Zde můžeme zadávat jak základ mocniny, tak exponent. Příklad:  $5^3 = 125$



## 13. Programování v FBD

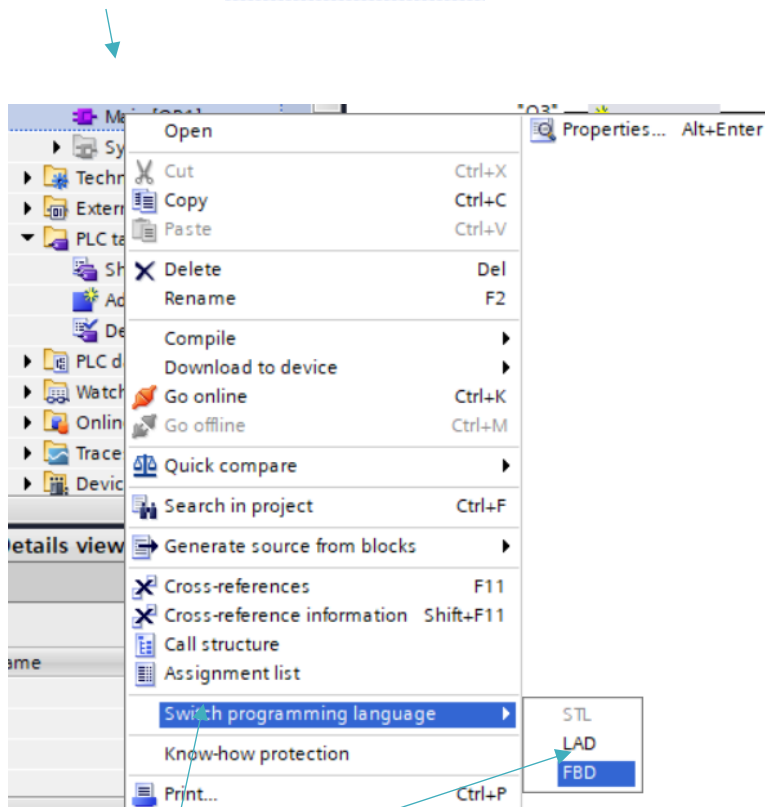
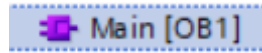
Zde je program již několikrát zmiňovaný pro spínání výstupu jedním tlačítkem a druhým pro vypnutí



Program V LAD můžeme převést do FBD a to dvěma způsoby:

A:

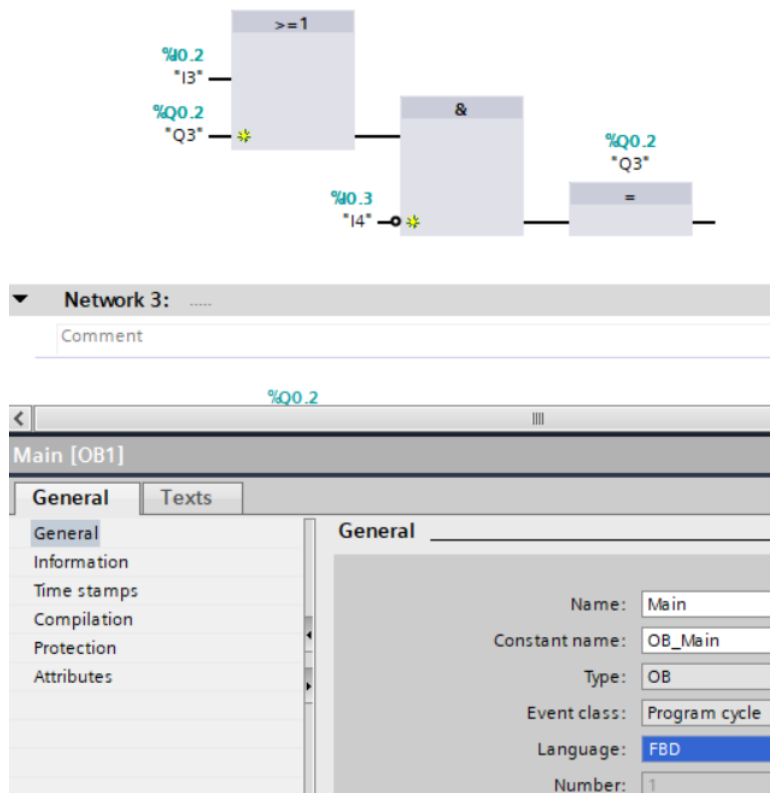
1. Klik LT na „Main“



2. Klik LT na „Switch programming Language“
3. Klik na „FBD“

B:

Druhý způsob je výběr typu programu v části Properties → General → Language, toto nastavení je pod programovací plochou, pokud není vidět musí se dělicí čára táhnout směrem nahoru.



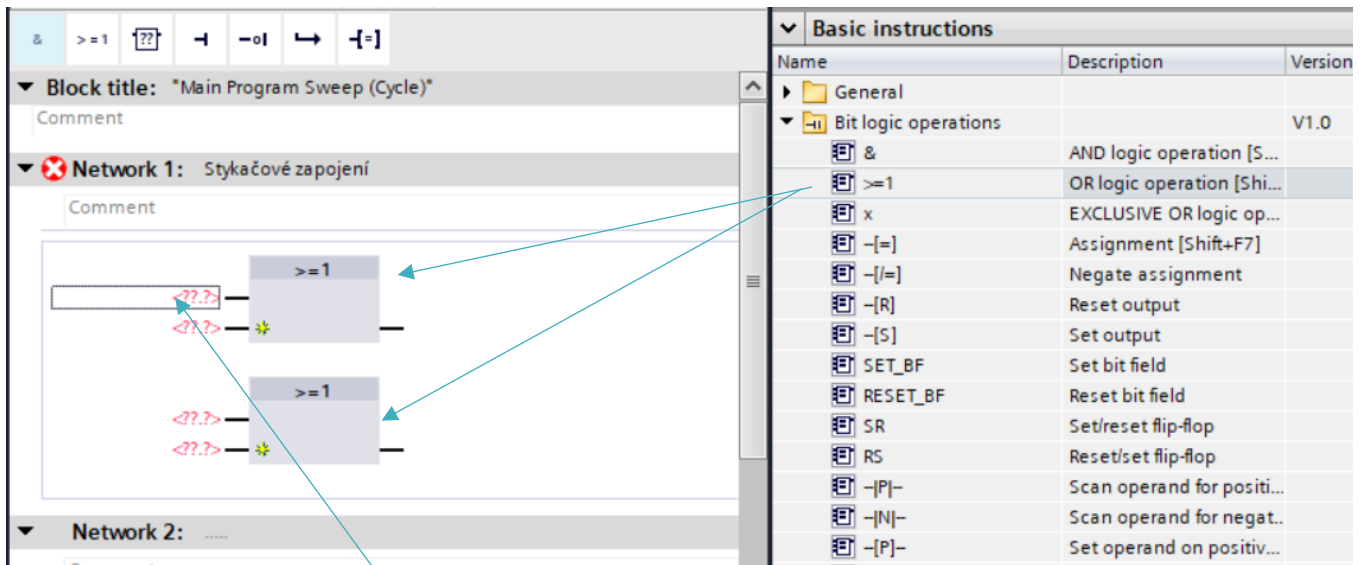
Stejným způsobem se přepne zpět do programu v LAD a stejně tak na počátku nastavíme typ programování.

Příklad: Jeden výstup budeme ovládat pomocí tlačítek a panelu HMI. Vstupy I1 a M1 budeme vstup zapínat a vstupy I2 a M2 budeme výstup Q1 vypínat.

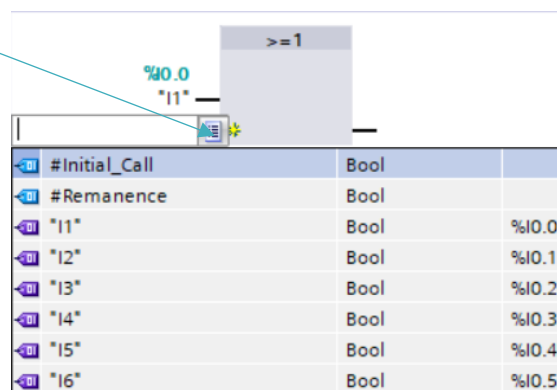
1. Zapišeme adresy proměnných

Default tag table			
	Name	Data type	Address
1	M1	Bool	%M0.0
2	Q1	Bool	%Q0.0
3	Q2	Bool	%Q0.1
4	M2	Bool	%M0.1

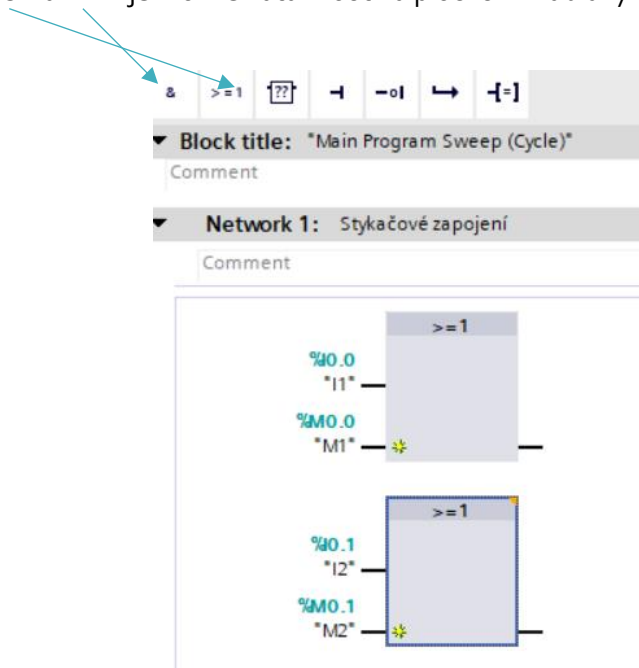
2. Na plochu natáhneme dva logické bloky OR.



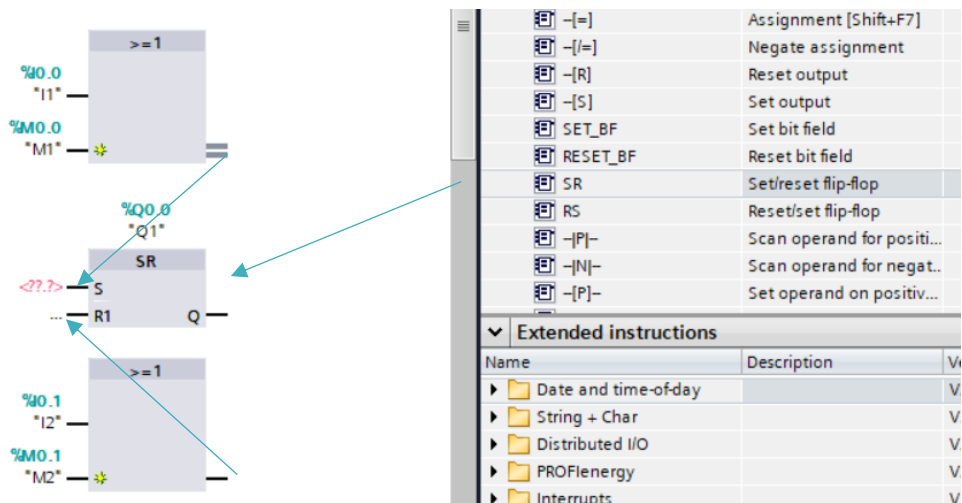
3. Vložíme vstupy – klik LT na otazníky a vybereme vstupy – otevře se obdélník a klikem na pravou značku se otevře nabídka „tag“, kterou v 1. bodě vytvořili.



Blok OR a AND je možné natáhnout na plochu i z nabídky nad programovací plochou.




4. Vložíme blok SR

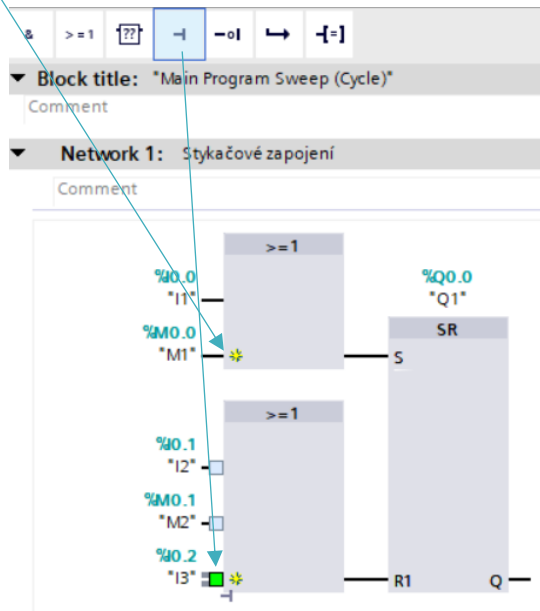


5. Propojíme bloky OR a Blok SR


Rozšíření vstupů bloků

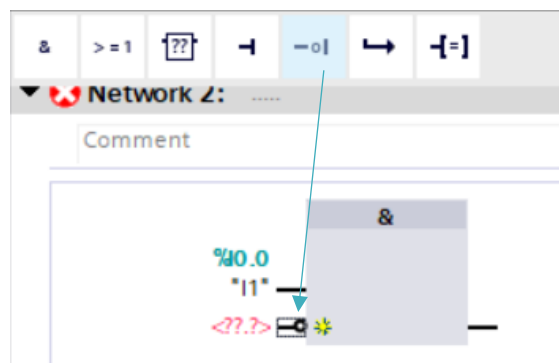


Vstupy bloků u nichž je v levém spodním rohu hvězdička můžeme přidávat. Přidání uskutečníme kliknutím na hvězdičku. Druhou možností je táhnout blok  na poslední vstup.



### Negace


Negaci vstupu, ale i výstupu provedeme tažením  značky na vstup nebo výstup, který chceme negovat.

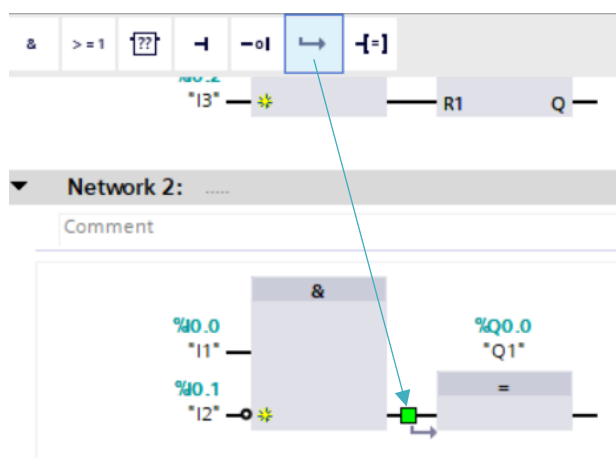


Negovaný vstup se změní na kružnici.

### Větvění programu

Příklad: Spínání dvou výstupů současně

Značku  táhneme na místo větvení



Na místo odbočení přidáme výstup

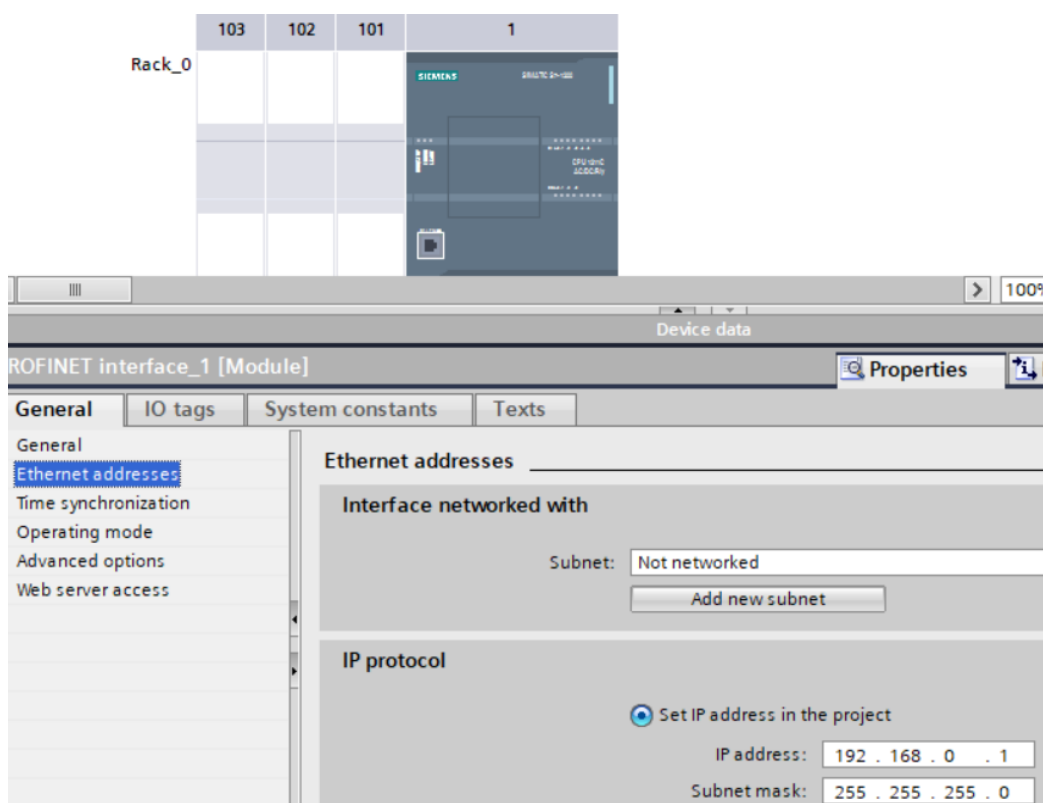


## 14. Analogové funkce

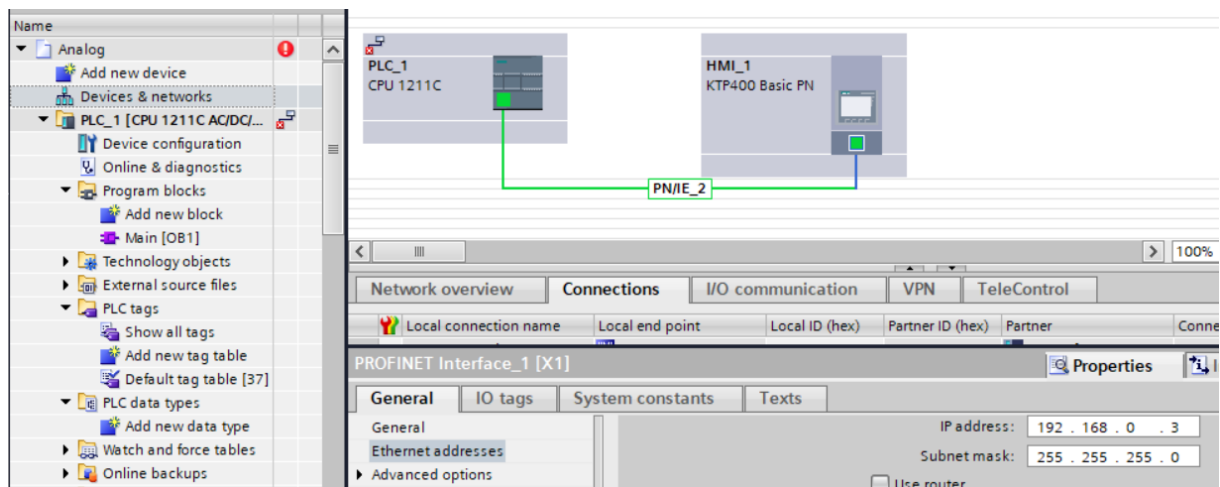
Veličiny jako je teplota, tlak, vlhkost, odpor (tenzometry) apod., mají spojitou hodnoty. Ty se pomocí externích převodníků, ty jsou vždy pro určitý rozsah měření např. teploty o až 200°C, nebo jsou univerzální a musí se nastavit, jak typ čidla, tak hodnota výstupu. Výstup z převodníku může být též jeden typ, nebo je volitelný. Standardní proudové výstupy jsou 0 – 20mA nebo 4 – 20mA a napěťové výstupy jsou 0-5V nebo 0 – 10V. Je-li výstup z převodníku 10V stačí základní modul S1200, pokud je výstup z převodníku jiný musí se do základního modulu S1200 vložit rozšiřující karta, která umožní výběr všech vstupů. Budeme-li uvedený rozsah převodníku 0°C až 200°C, představuje 1°C hodnotu 0,05V ( $10/200=0,05$ ). Pro 50°C to bude 2,5V ( $0,05 \times 50=2,5$ ). Analogová hodnota z převodníku se v S1200 převádí na digitální v rozsahu 0 až 27648. Tyto mezní hodnoty se nastavují jako min a max ve vstupním bloku NORM X.

Nastavení adres analogových vstupů v základním přístroji S 1200. Z

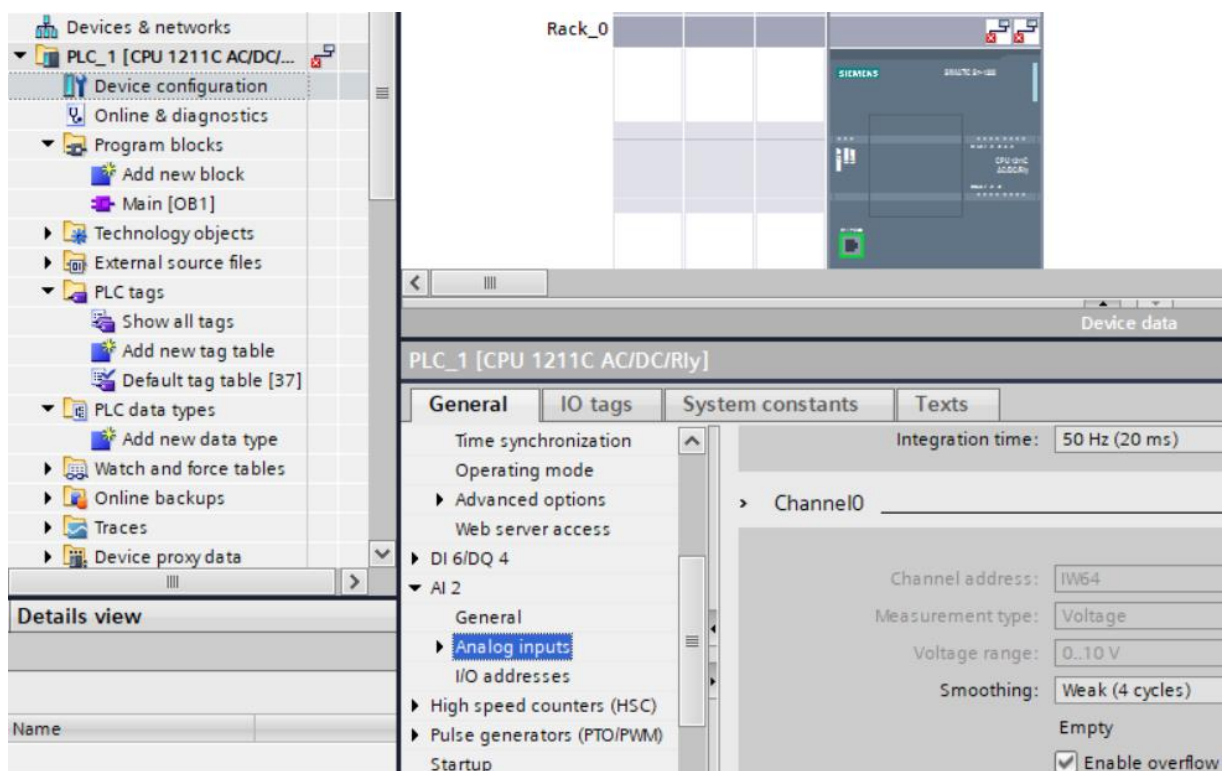
Zjištění adresy přístroje. Pokud budeme přidávat další přístroj např. panel HMI musíme mít na přístrojích různé adresy. Adresu najdeme po kliknutí na výstup ethernetového výstupu (Properties → General → Ethernet addresses).



Vložené přístroje a jejich propojení zjistíme po kliknutí na ikonu Device&networks. IP adresu panelu HMI zjistíme po kliknutí na ethernetový vstup panelu HMI a dále kliknutím na Properties → General → Ethernet addresses . Adresy jsou rozdílné, pokud ne tak ji přepíšeme.

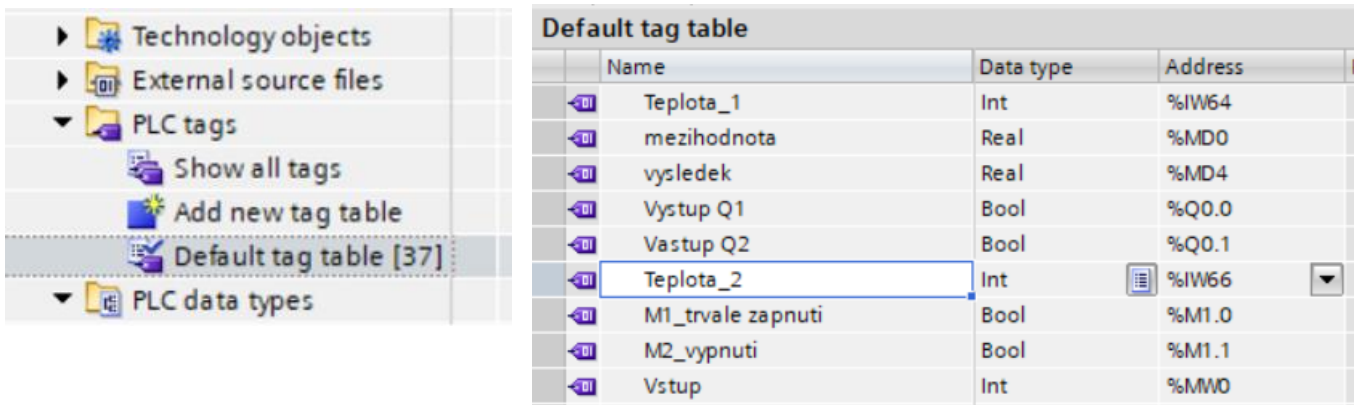


Klikneme na zobrazení bloku. V okně General → Analog inputs vidíme, že nemůžeme měnit analogové vstupy. V tomto případě můžeme použít pouze převodník 0-10V.



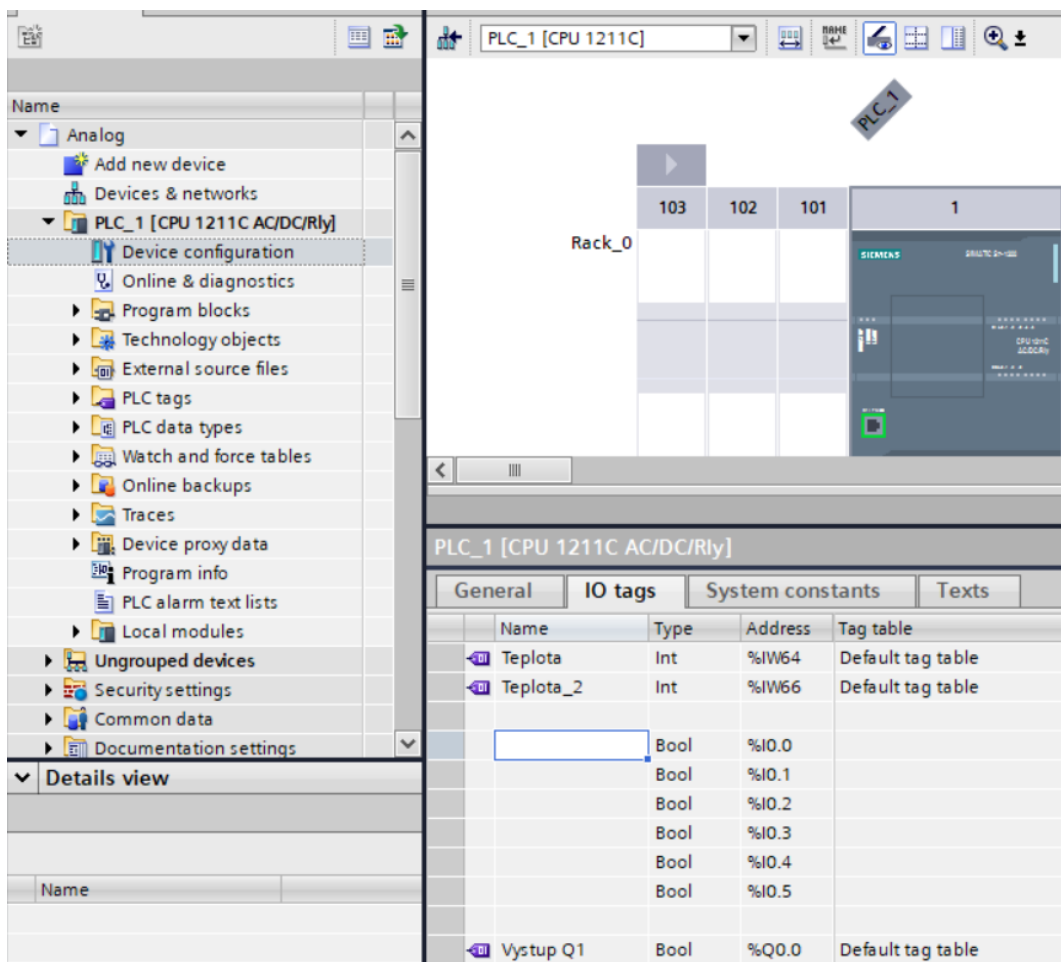
## Nastavení adres proměnných

Adresy můžeme nastavit PLC tags → Default tag table. Vstupy jsou dva a mají pevné adresy **IW64** a **IW66**, tyto adresy nelze zaměnit za jiné.



Default tag table			
	Name	Data type	Address
	Teplota_1	Int	%IW64
	mezihodnota	Real	%MD0
	vysledek	Real	%MD4
	Vystup Q1	Bool	%Q0.0
	Vastup Q2	Bool	%Q0.1
	Teplota_2	Int	%IW66
	M1_trvale zapnuti	Bool	%M1.0
	M2_vypnuti	Bool	%M1.1
	Vstup	Int	%MWO

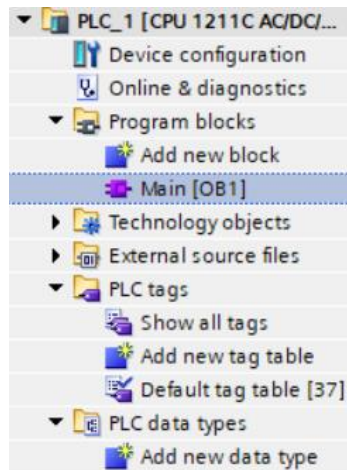
Adresu analogových vstupů vidíme po otevření „Device configuration“ po kliknutí na IO tags.



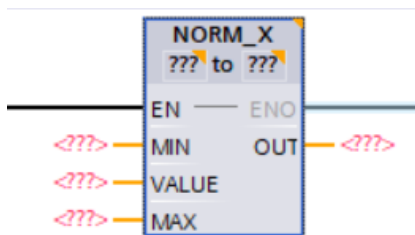
PLC_1 [CPU 1211C AC/DC/Rly]				
General	IO tags	System constants	Texts	
	Name	Type	Address	Tag table
	Teplota	Int	%IW64	Default tag table
	Teplota_2	Int	%IW66	Default tag table
		Bool	%IO.0	
		Bool	%IO.1	
		Bool	%IO.2	
		Bool	%IO.3	
		Bool	%IO.4	
		Bool	%IO.5	
	Vystup Q1	Bool	%Q0.0	Default tag table

## Vytvoření programu

Přepneme se na programovací plochu „Main“

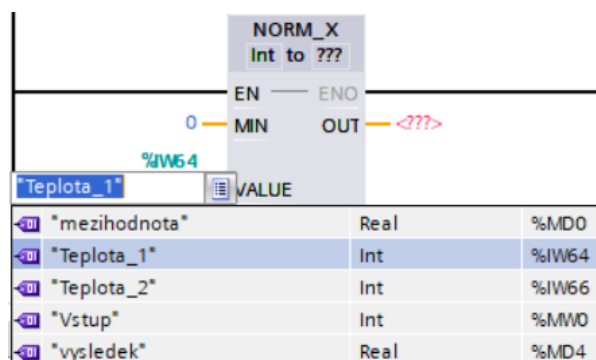


1. Na řádek na programovací ploše vložíme blok **NORM\_X**.



▶	General	
▶	-1 Bit logic operations	
▶	⌚ Timer operations	
▶	+1 Counter operations	
▶	< Comparator operations	
▶	± Math functions	
▶	➡ Move operations	
▼	↔ Conversion operations	
☑	CONVERT	Convert value
☑	ROUND	Round numerical value
☑	CEIL	Generate next higher integer fro...
☑	FLOOR	Generate next lower integer fro...
☑	TRUNC	Truncate numerical value
☑	SCALE_X	Scale
☑	NORM_X	Normalize

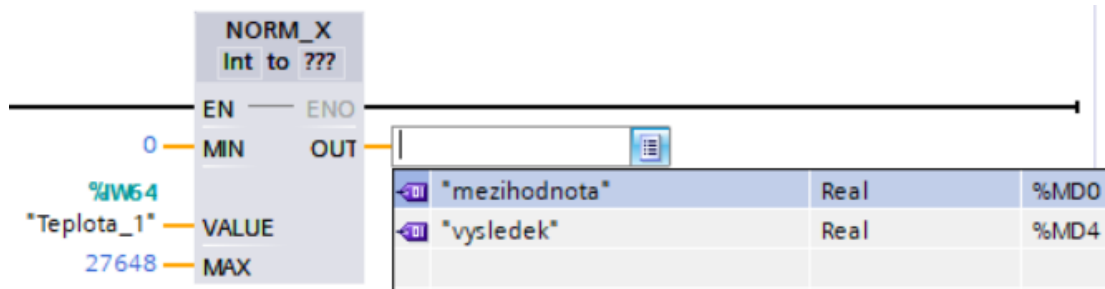
2. Na vstup MIN zapíšeme hodnotu „0“, Na vstup „VALUE“ vložíme adresu analogového vstupu.



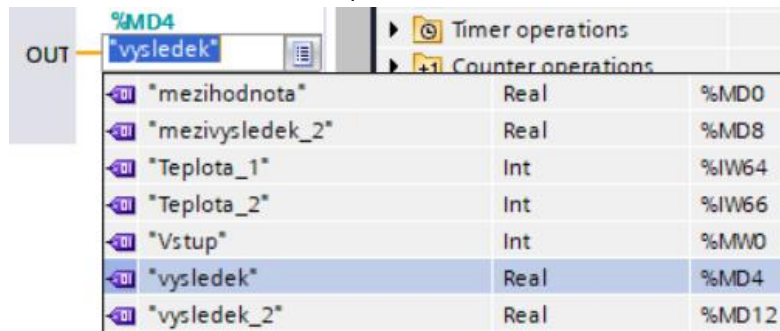
3. Na vstup MAX zapíšeme hodnotu „27648“
4. Na výstup OUT vložíme adresu kterou jsme pojmenovali „mezihodnota“. Výsledek OUT bude nabývat hodnotu v intervalu 0 – 1.

$$\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN})$$

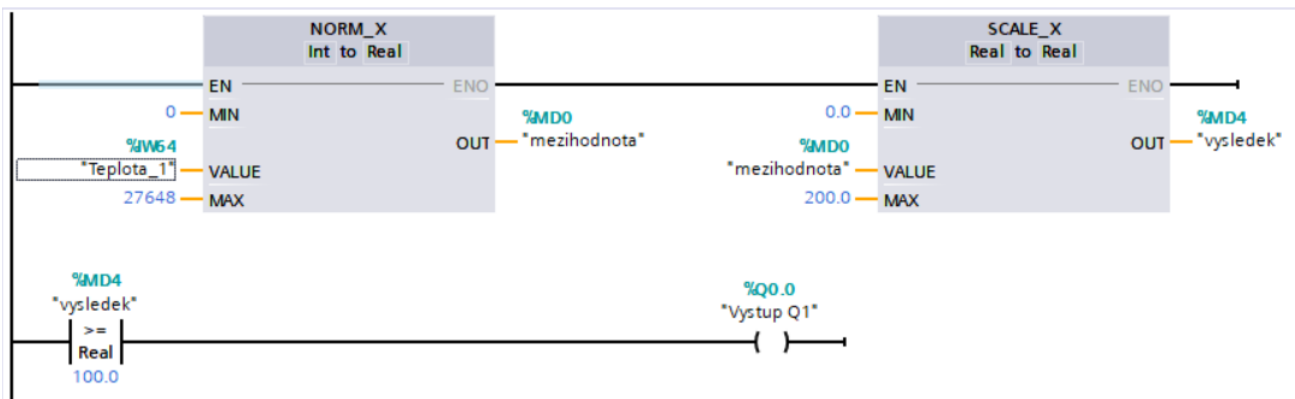
Př.: Hodnota VALUE je 13824, potom  $(13824 - 0) / (27648 - 0) = 0,5$



5. Připojíme blok SCALE\_X
6. Na vstup MIN zapíšeme minimální hodnotu rozsahu na převodníku.
7. Na vstup VALUE vybereme adresu shodnou s výstupem bloku NORM\_X „mezihodnota“
8. Na vstup MAX zapíšeme maximální hodnotu rozsahu převodníku.
9. Na výstup OUT vložíme adresu výstupu.



Výsledné zapojení bloků



Ve druhém řádku je příklad doplněn o komparátor, který sepne výstup, je-li hodnota výstupu větší nebo rovna než hodnota zadaná ve spodní části bloku (100)

Hodnota na výstupu bloku SCALE\_X se vypočte:  $OUT = [VALUE * (MAX - MIN)] + MIN$

Př: Rozsah měření teploty je 0°C až 200°C, vstup je 0,5.  $OUT = 0,5 * (200 - 0) + 0 = 100$  °C

V následující tabulce je přepočten vstup a výstup pro rozsah teploty od -50°C do 100°C

Příklad výpočtu jedné teploty z tabulky.

$$OUT = (VALUE - MIN) / (MAX - MIN)$$

$$(8294 - 0) / (27648 - 0) = 0,3$$

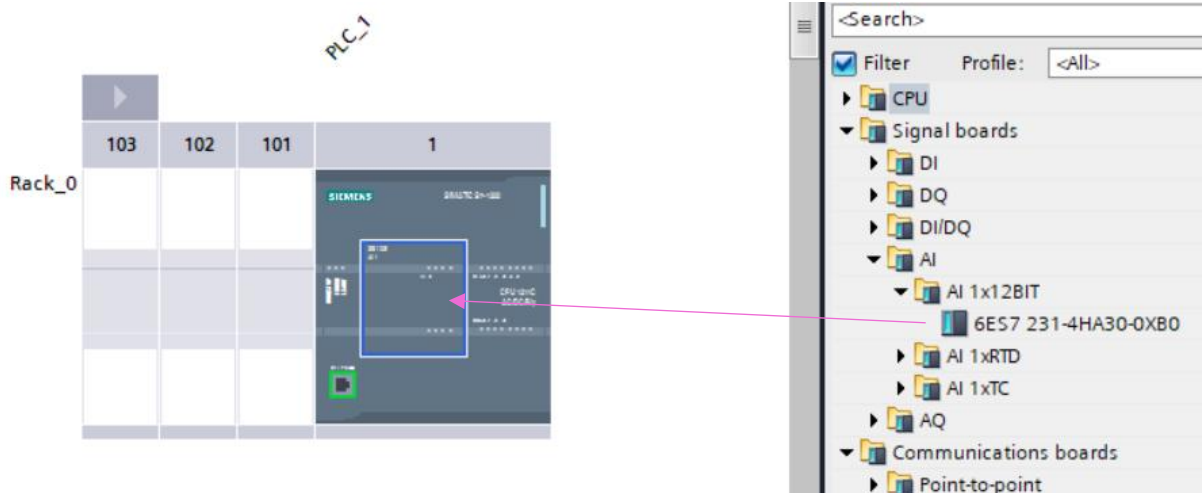
$$OUT = [VALUE * (MAX - MIN)] + MIN$$

$$OUT = 0,3 * [100 - (-50)] + [-50] = -50$$

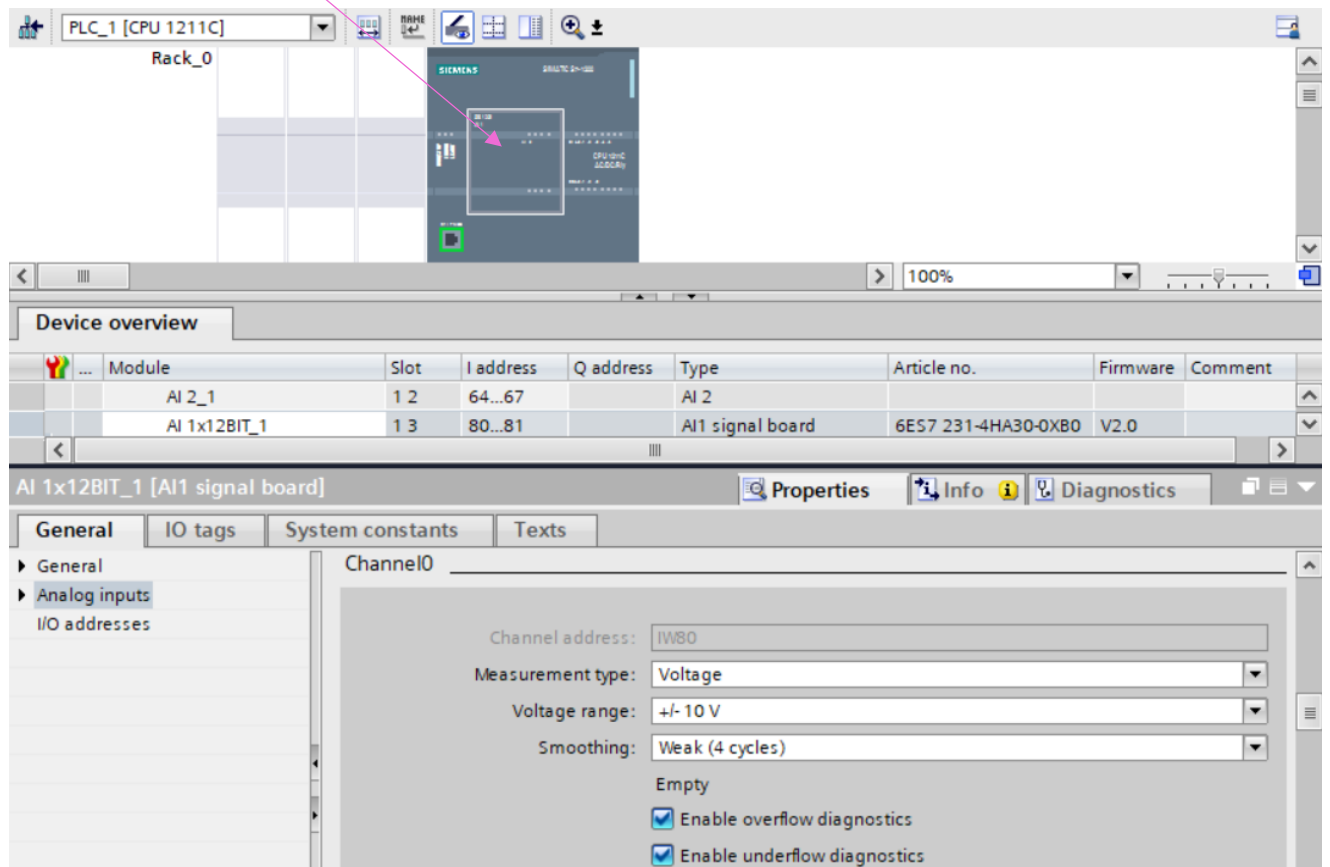
vstup Value NORM_X	mezihodnota	výsledná teplota SCALE_X
0	0	-50
2 765	0,1	-35
5 530	0,2	-20
8 294	0,3	-5
11 059	0,4	10
13 824	0,5	25
16 589	0,6	40
19 354	0,7	55
22 118	0,8	70
24 883	0,9	85
27 648	1	100

### Přidání analogové karty

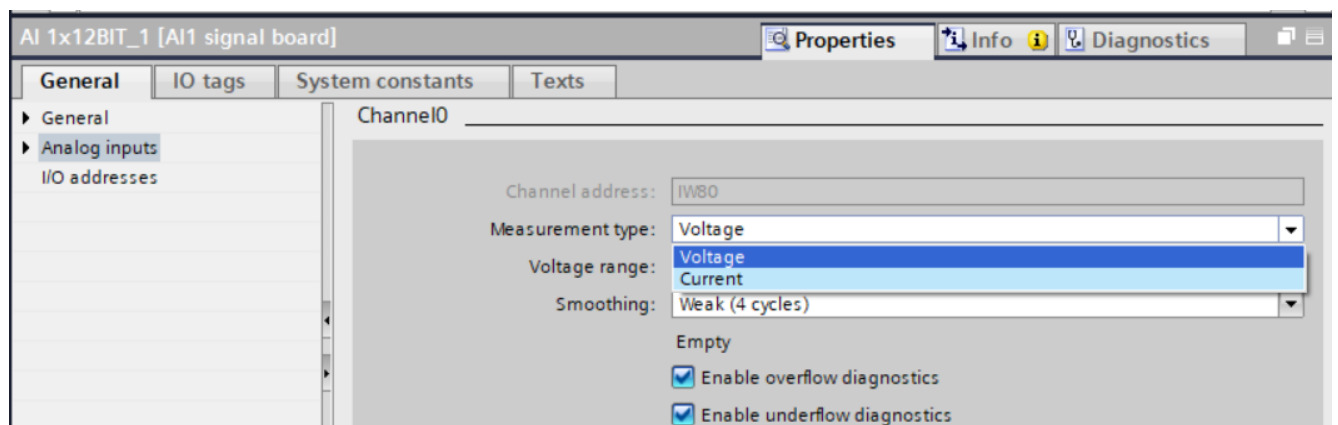
Typ, přístroje, který byl vybrán nemá možnost pro rozšíření dalšími spínacími bloky, ale do přístroje je možné fyzicky vložit kartu s analogovým vstupem. V programu tuto kartu vložíme táhnutím myši ikony s názvem karty „6ES7 231-4AH30-0XB0“ do místa ve středu přístroje.



Klikneme na blok karty. Po otevření panelu, kliknutím na Properties → General se otevře okno pro nastavení převodníku.



Zvolíme podle typu převodníku v okně „Measurement type“ zda se jedná o proudový nebo napěťový převodník (Voltage – napěťový, Current - proudový)



V okně „Voltage range“ zvolíme napěťový rozsah. Nejčastější bývá o - 10V. Volba je opět závislá na typu převodníku,

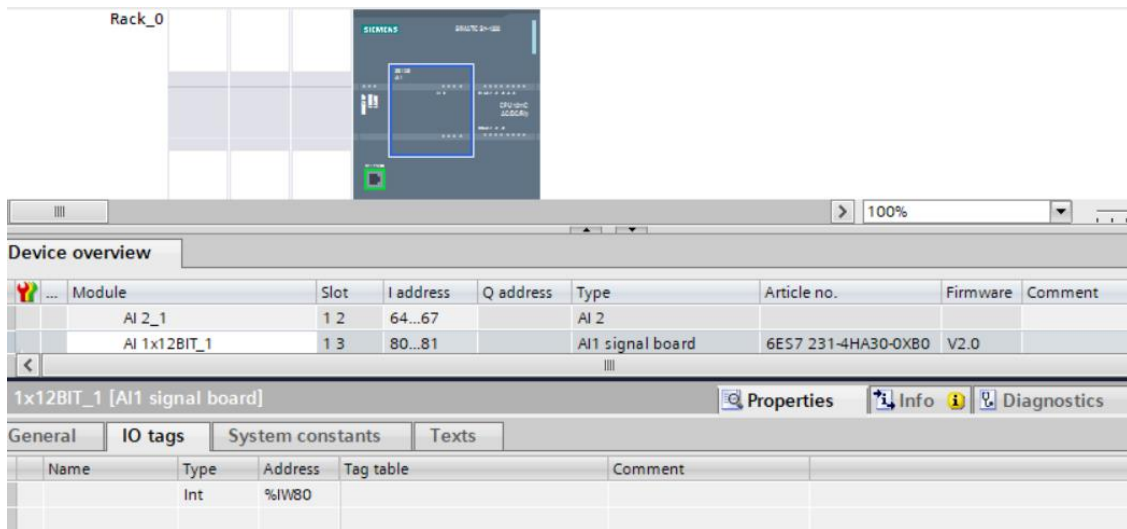
Channel address: IW80  
Measurement type: Voltage  
Voltage range: +/- 10 V  
Smoothing: +/- 10 V  
 Enable overflow diagnostics  
 Enable underflow diagnostics

U proudového typu volba není, zde musíme použít proudový převodník o – 20mA.

Channel address: IW80  
Measurement type: Current  
Current range: 0..20 mA  
Smoothing: Weak (4 cycles)  
Empty  
 Enable overflow diagnostics  
 Enable underflow diagnostics

Zapsání adresy analogového vstupu

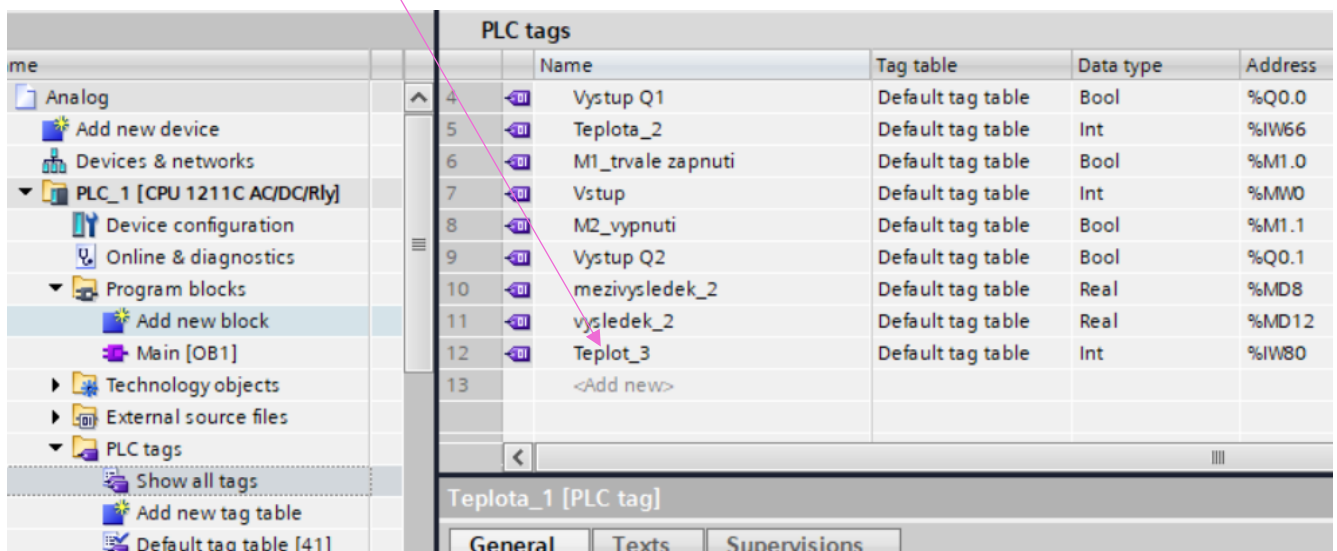
Klikneme na ikonu IO tags. Zde se nám zobrazí adresa **IW80**. Stačí do sloupce napsat název adresy. Adresu můžeme zadat i „PLC tags“ nebo v „Data block“, ale adresa musí být **IW80** a datový typ **Int**



Adresa po zapsání názvu okně „Name“

Name	Type	Address	Tag table
Teplot_3	Int	%IW80	Default tag table

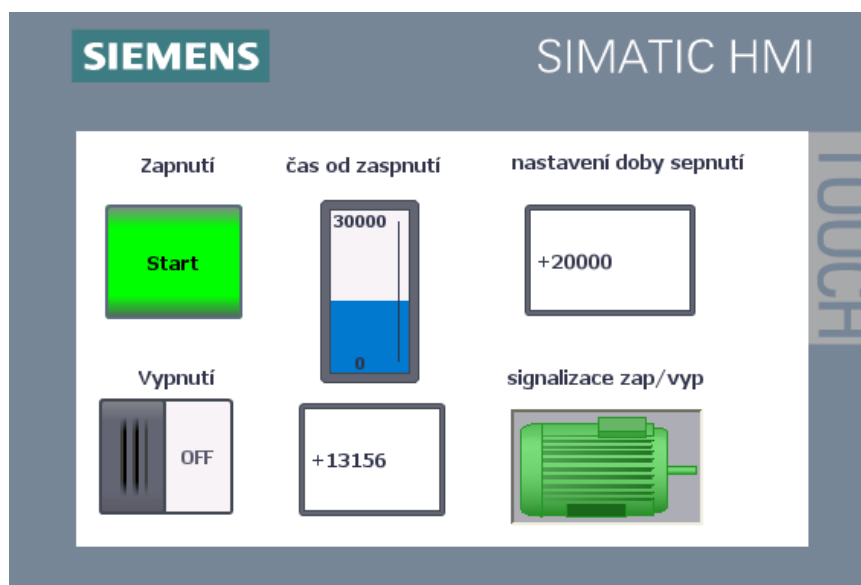
Kliknutím na PLC → Show all tags se zobrazí tabulka se všemi adresami, které jsme zapsali, zde můžeme vidět i adresu zapsanou „IO tags“



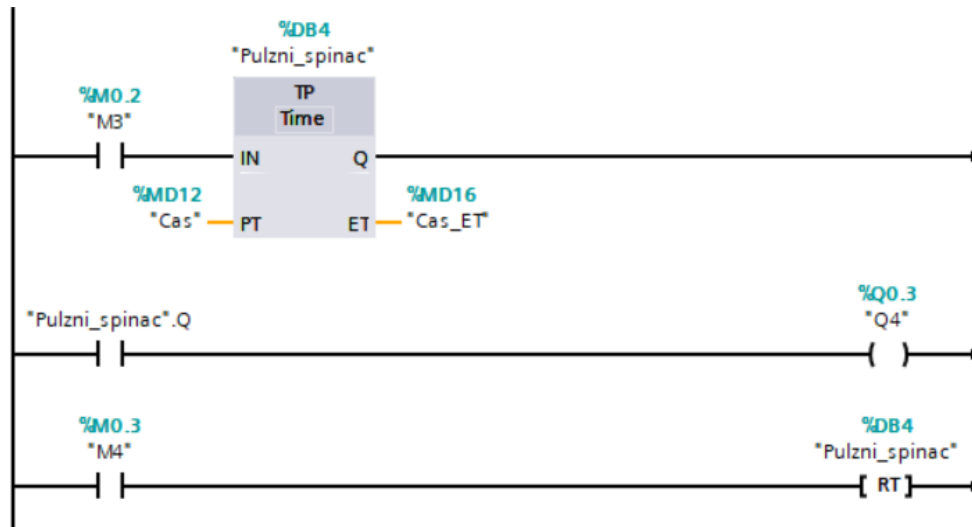
## 15. Propojení s panelem HMI

Vkládání bloků na plochu, jejich nastavení a grafická úprava je podrobně popsána v části samostatného návodu „LOGO! Krok za krokem II“.

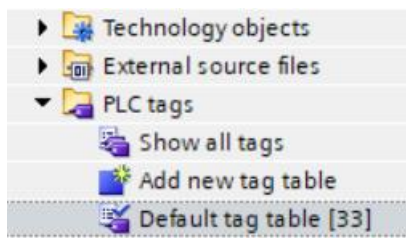
Na panelu zobrazíme tlačítko, vypínač, grafické (sloupcový graf“ a číselné zobrazení času, nastavení času a grafické znázornění sepnutí (motor).



Program, který bude ovládán z panelu HMI: Sepnutím spínače M<sub>3</sub> na panelu HMI tlačítko „Start“ se sepne časovač TP. Doba sepnutí výstupu je dána vstupem PT. Čas sepnutí se nastaví kliknutím na okno „nastavení doby sepnutí“. Sepnutím Q časovače TP se sepne spínací kontakt „Pulzní spínač na druhém řádku a výstup „Q<sub>4</sub>“. Sepnutím výstupu Q se začne na panelu zobrazovat aktuální čas od sepnutí (výstup ET), zvyšující čas se začne zobrazovat i graficky. Sepnutí se také zobrazí změnou barvy motoru (zelená zapnuto, šedá vypnuto). Spínačem M<sub>4</sub> na panelu HMI „Vypnutí“ se zastaví časování, před jeho dokončení.



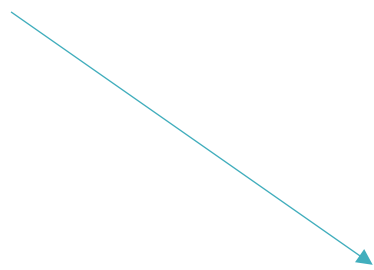
Proměnné nastavíme v PLC tags → Default tag table, můžeme je nastavit i v „Data block“. Musí však dávat pozor, kde je hledat.

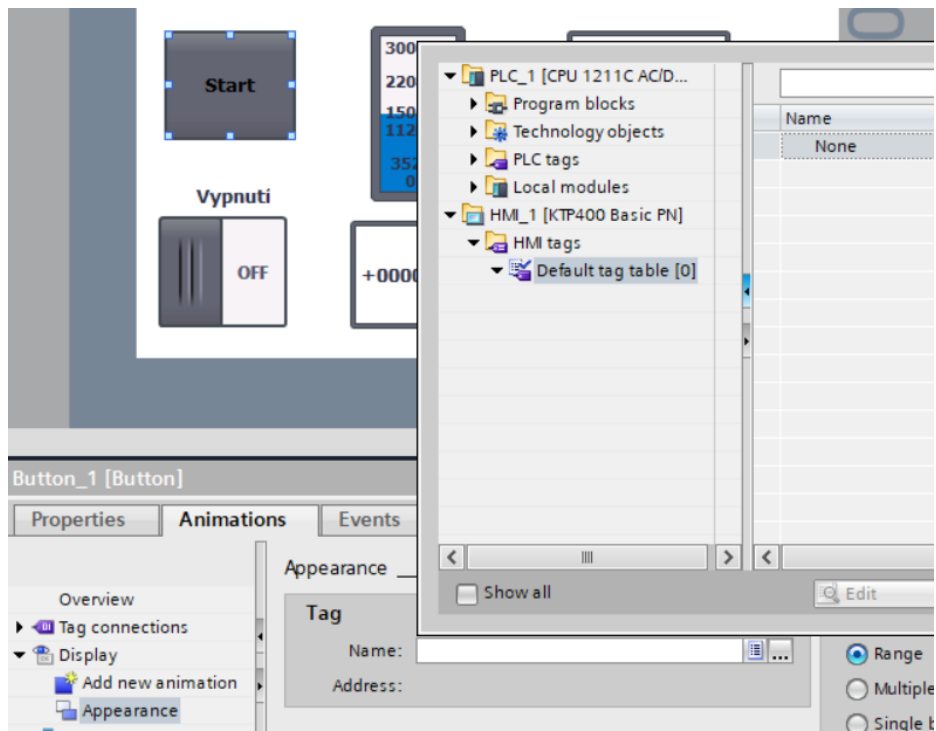


Default tag table			
	Name	Data type	Address
DI	M3	Bool	%M0.2
DI	M4	Bool	%M0.3
DI	Q4	Bool	%Q0.3
DI	Cas	Time	%MD12
DI	Cas_ET	Time	%MD16

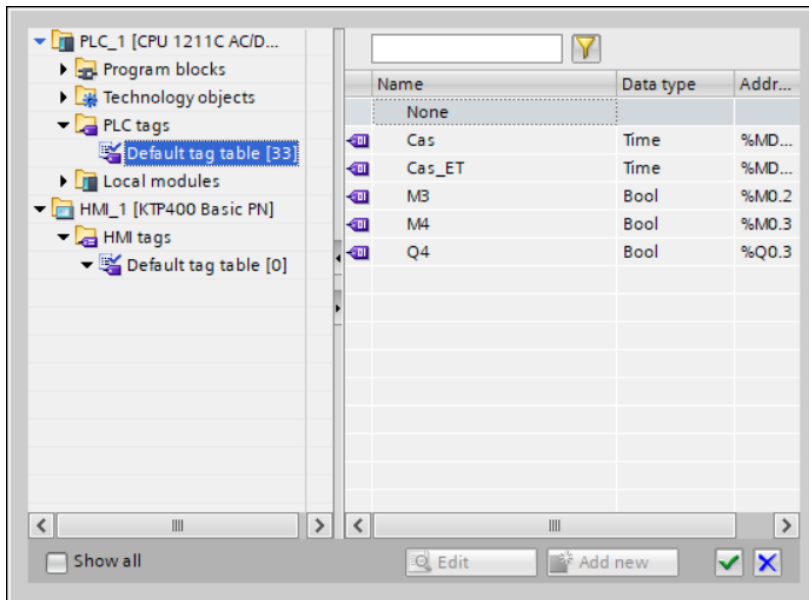
Spínací kontakt M3 má nastavení pro „Animations → Appearance“ pro střídání barvy tlačítka a Events pro spínání.

Kliknutím na rozbalovací ikonu Tag máme prázdnou tabulku.

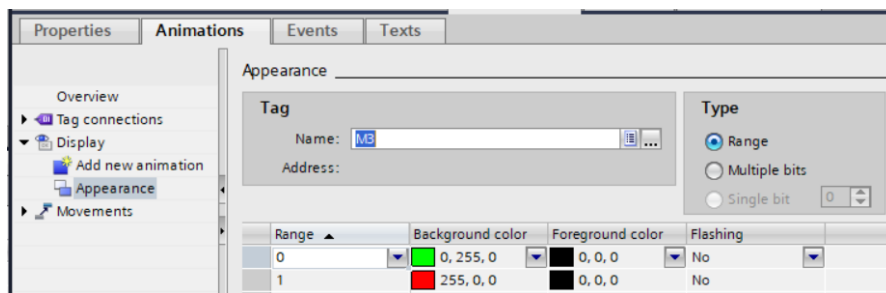




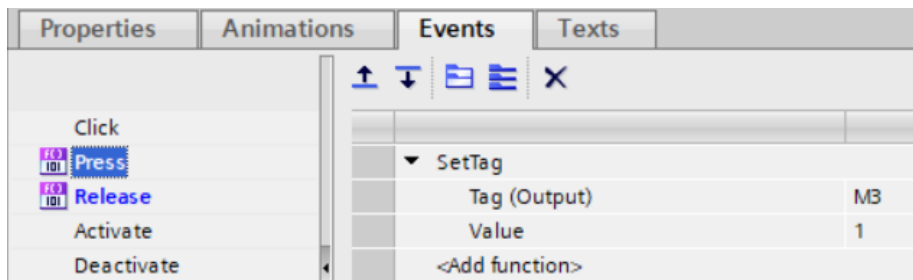
Musíme přejít do PLC tags → Default tag table a tam vybereme M3.



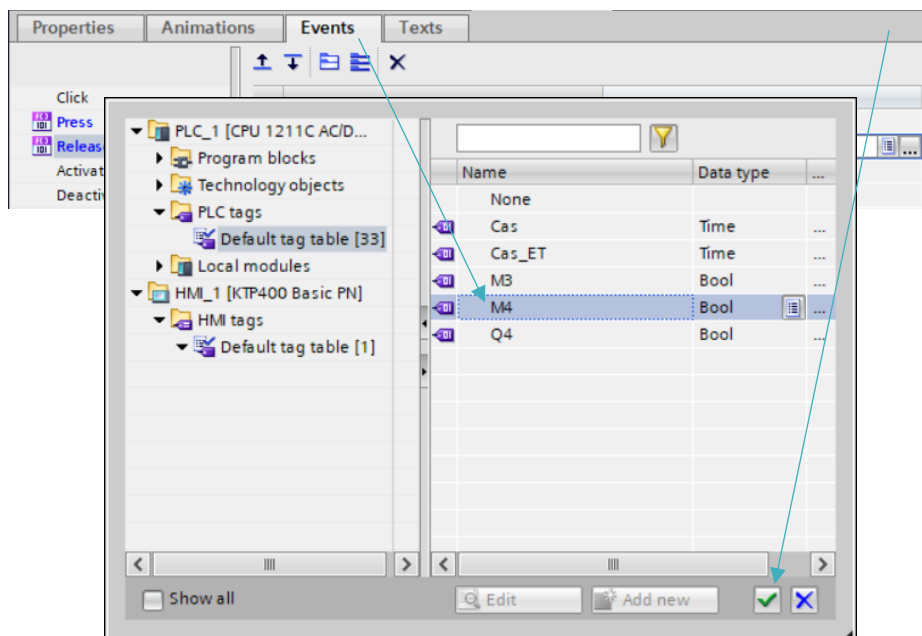
Nastavení pro střídání barev na tlačítku



Nastavení „Evens“ pro sepnutí tlačítka „Press“

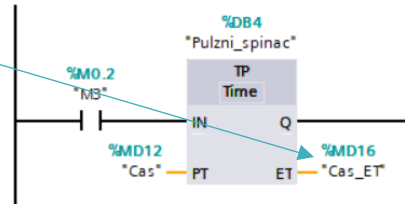


Nastavení „Evens“ pro rozpojení tlačítka „Release“.



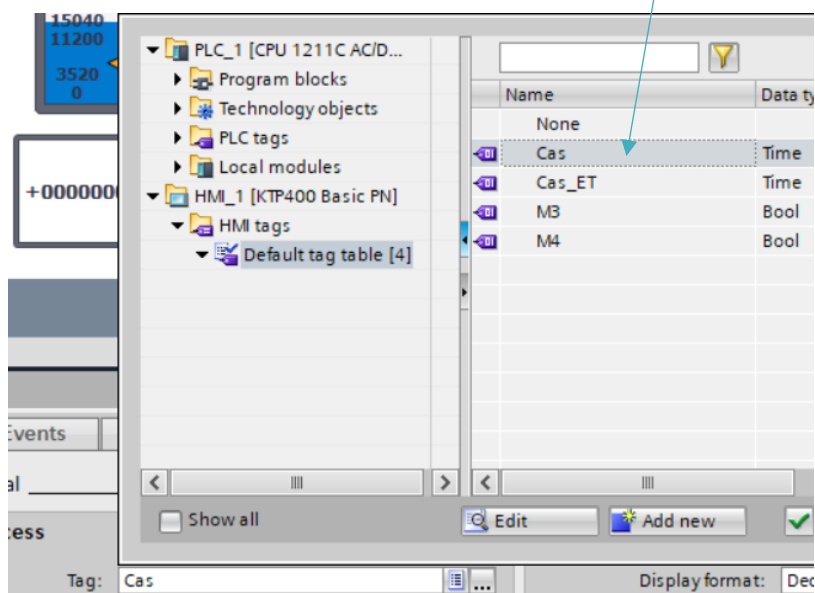
Nastavení adresy spínače "Vypnutí" M4. Nesmí se zapomenout na kliknutí na zelené zaškrtnací políčko.

Nastavení adresy grafu, je výstup ET z časovače „Cas\_ET“.



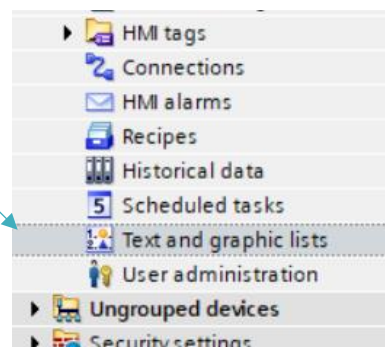
Nastavení adresy pro zobrazování doby sepnutí, je stejné jako v předchozím případě výstup ET z časovače „Cas ET“.

Adresa pro nastavení doby sepnutí je adresa vstupu PT do časovače TP "CAS"

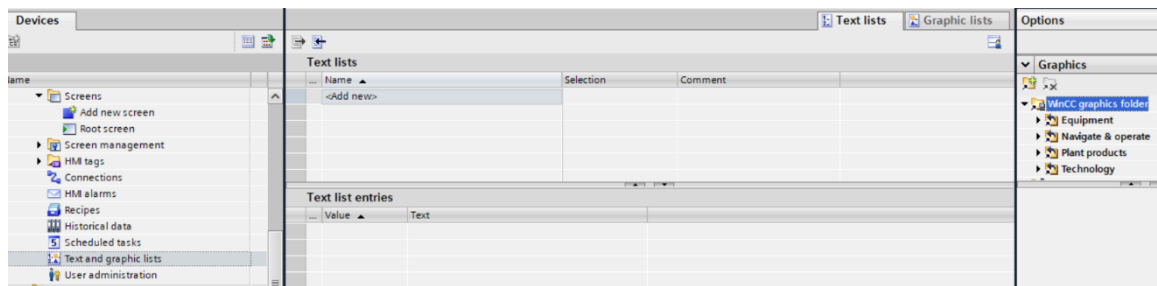


Vložení obrázku, u něhož se bude měnit zbarvení podle stavu zapnuto/ vypnuto se provede následovně:

1. Dvojklik LT na ikonu „Text and graphic list“

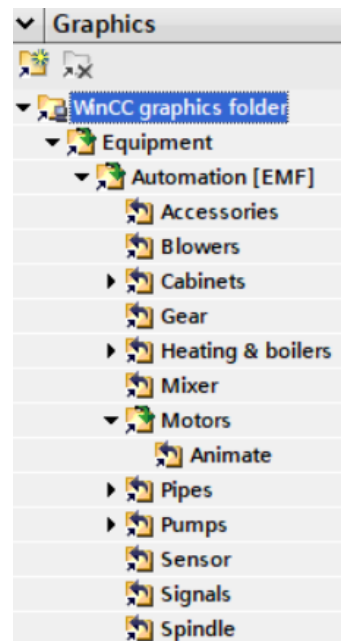


Otevře se obrazovka s tabulkou a v pravé části s nabídkou grafických prvků, ze které si vybereme požadovaný obrázek.



2. Kliknutím na rozbalovací šipky vybereme požadovaný obrázku – zde vybereme „Motors“ .

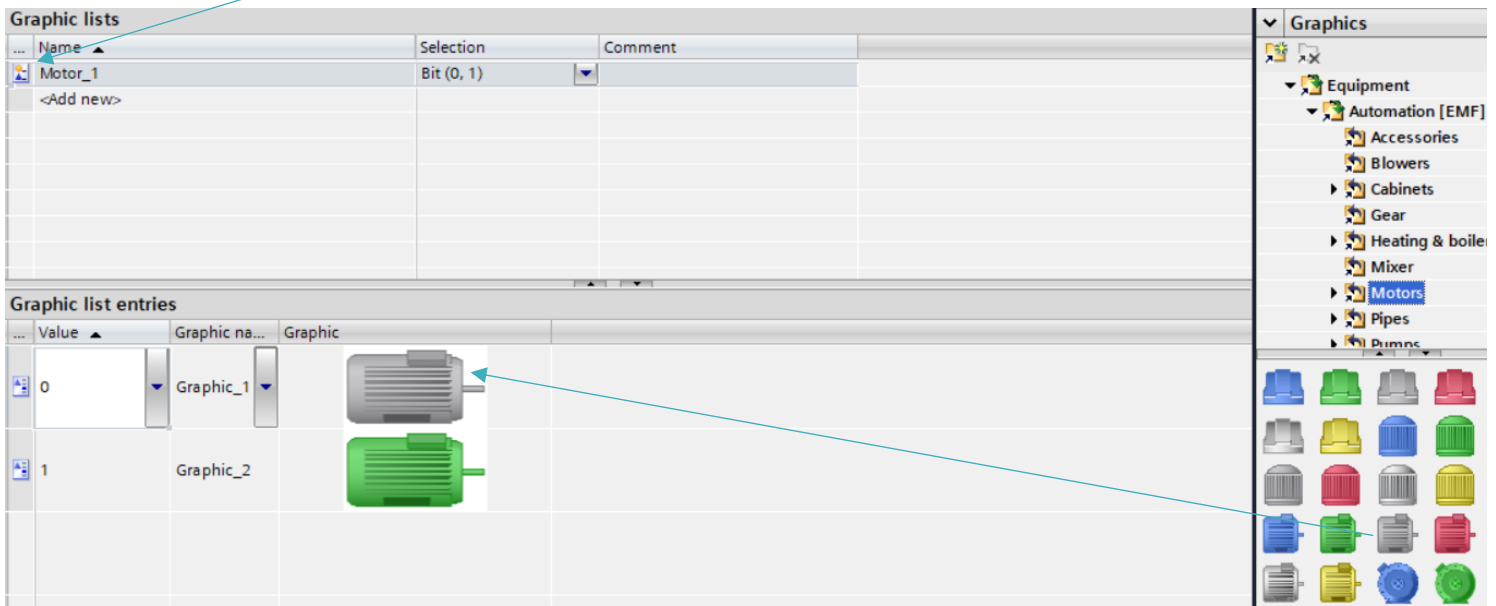
Kliknutím na Motors se zobrazí nabídka motorů



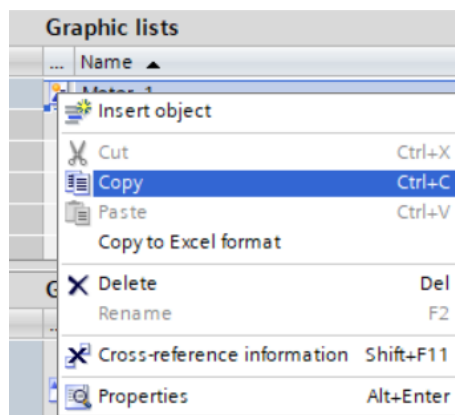
typ

3. Klik na ikonu Graphic lists - otevře se tabulka
4. Zapišeme název např. „Motor\_1“
5. Do tabulky „Graphic list entris“ přetáhneme obrázky motorů

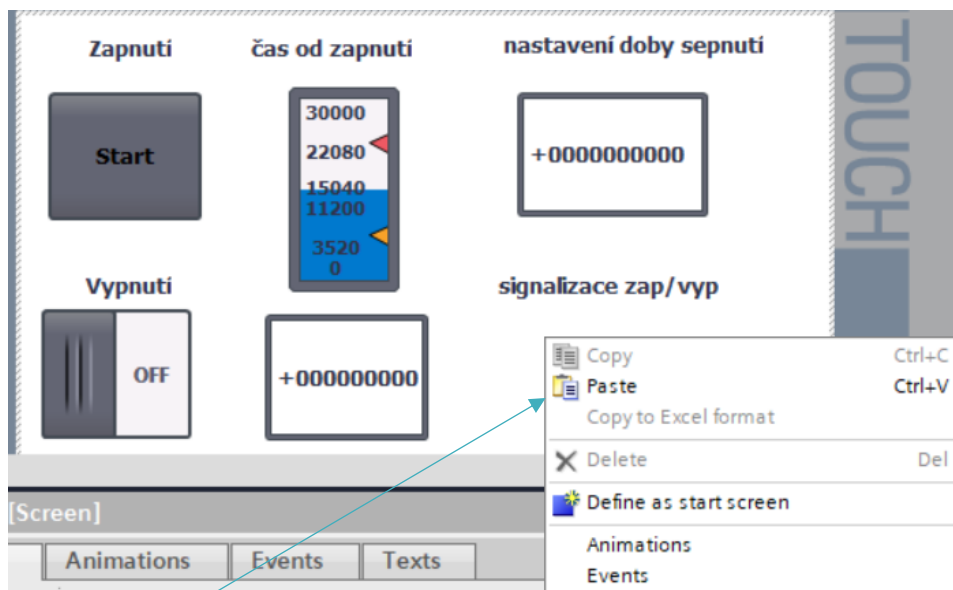
6. Klik PT na ikonu u Motor\_1



7. Klik LT na Copy



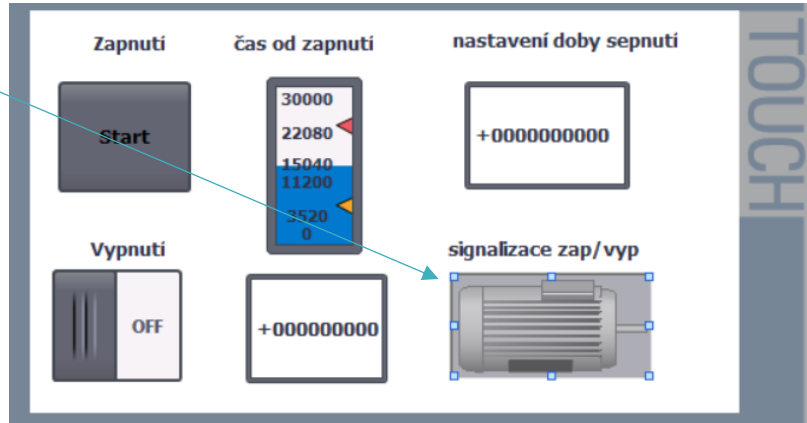
8. Otevře se obrazovka panelu HMI



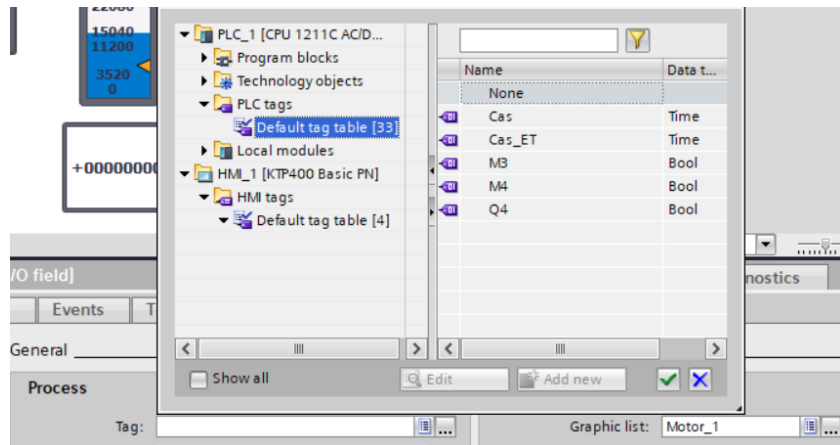
9. Klik PT do panelu

10. Klik LT na „Paste“ – vloží se obrázek motoru

11. Klik LT na obrázek motoru



12. Nastaví se adresa motoru - výstupu „Q4“ .



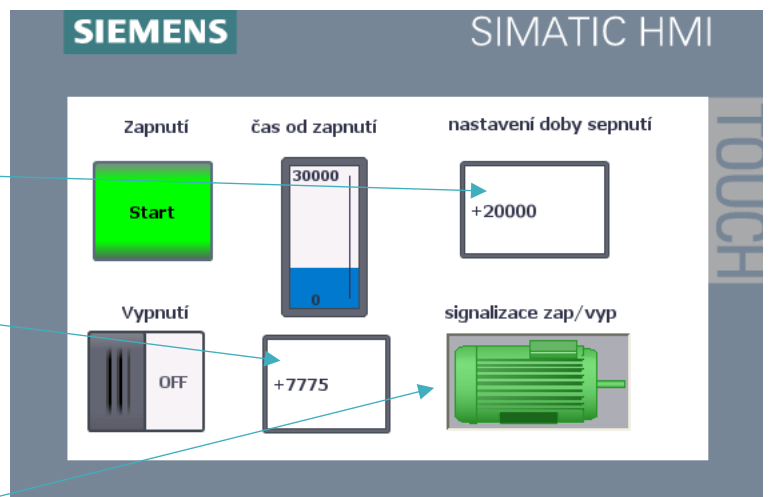
Ovládání z naprogramovaného panelu HMI:

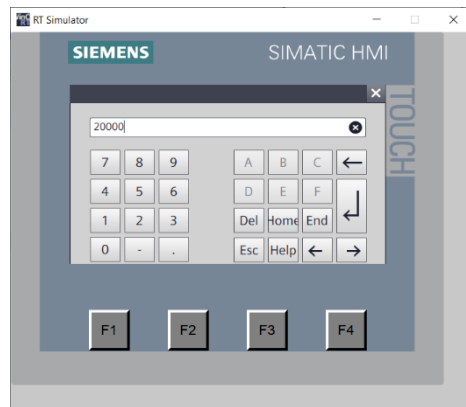
Kliknutím do okna „nastavení doby sepnutí“ se otevře klávesnice na níž nastavíme čas v ms.

V okně čas od zapnutí se zobrazuje čas v ms, který běží od stisku tlačítka Start.

Se zvyšujícím se časem stoupá modrý sloupec v grafu.

Po stisku Start se zobrazí zeleně motor





## 16. PŘÍKLADY

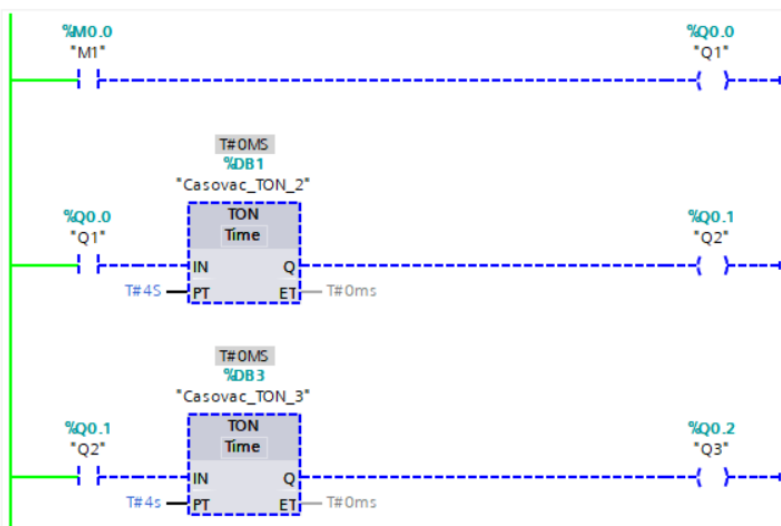
Z důvodu simulace jsou pro vstupy a výstupy použity paměti „M“, v praxi by se ve většině použili vstupy „I“ a výstupy „Q“.

### a) Postupné spínání

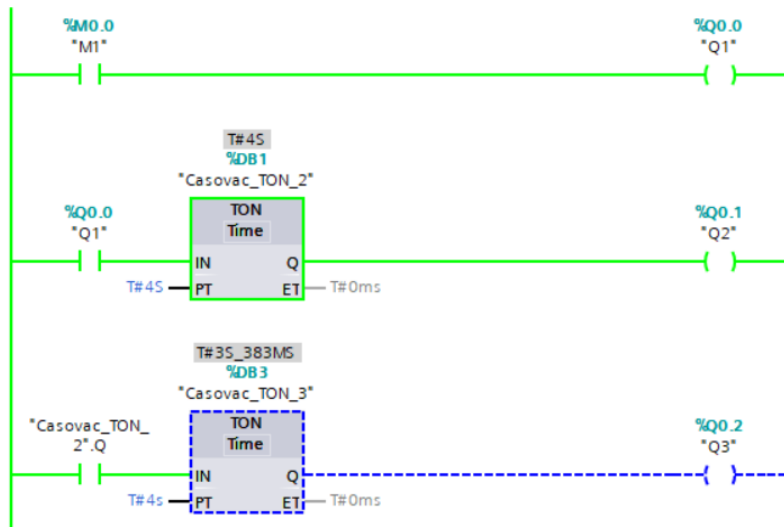
Sepnutím spínače M1 se sepne výstup Q1, po 4s se sepne výstup Q2 a po dalších 4s se sepne výstup Q3. Za 8s budou všechny výstupy sepnuté. Jejich vypnutí se provede vypnutím spínače M1, a to najednou.

Tento způsob spouštění můžeme použít u strojních linek, kde první se spustí stroj na výstupu a poslední na vstupu např. dopravník pro přísun materiálu.

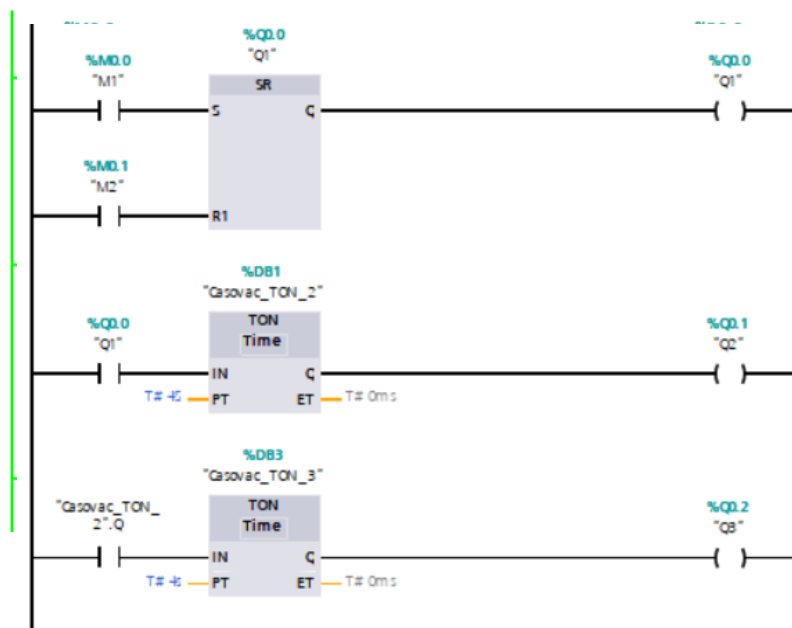
Program při stavu vypnutí všech výstupů.



Program ve stavu kdy je sepnutý výstup Q1 a Q2 a probíhá časování bloku „Casovac\_TON\_3“



Budeme-li chtít použít ovládání tlačítka, jedním zapnout a druhým vypnout, přidáme do prvního řádku blok SR. „M1“ zapne postupné zapínání výstupů a „M2“ všechny výstupy najednou vypne.



Sepnutím spínače M1 se sepne výstup Q1, po 4s se sepne výstup Q2 a po dalších 4s se sepne výstup Q3. Za 8s budou všechny výstupy sepnuté. Jejich vypnutí se provede vypnutím spínače M2, a to najednou.

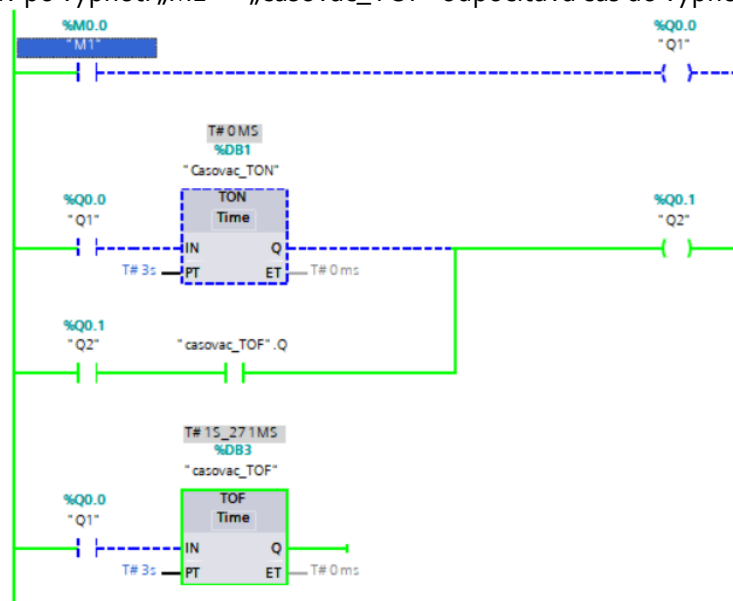
Tento způsob spouštění můžeme použít u strojních linek, kde první se spustí stroj na výstupu a poslední na vstupu např. dopravník pro přisun materiálu

### b) Postupné spínání a vypínání – první se zapíná i vypíná Q1

Při spuštění  $M1 = 1$ , se okamžitě sepne výstup  $Q1$ , po nastavené době se sepne výstup  $Q2$ . Při vypnutí  $M1 = 0$  se vypne první  $Q1$  a po nastaveném čase  $Q2$ .

V programu jsou použity dva časovače TON- zpožděné zapnutí a TOF zpožděné vypnutí. Při sepnutí „ $M1$ “ se spustí časování „TON“ a současně i vstup do „TOF“. Vypnutím „ $M1$ “ se vypne „ $Q1$ “, a začne odpočítávat čas „TOF“. „ $Q2$ “ zůstává sepnuté a jeho sepnutí zajišťují kontakty „TOF“ a „ $Q2$ “ ve třetím řádku. Po dokončení časování se kontakty zpožděného vypnutí rozpojí a vypne se výstup „ $Q2$ “.

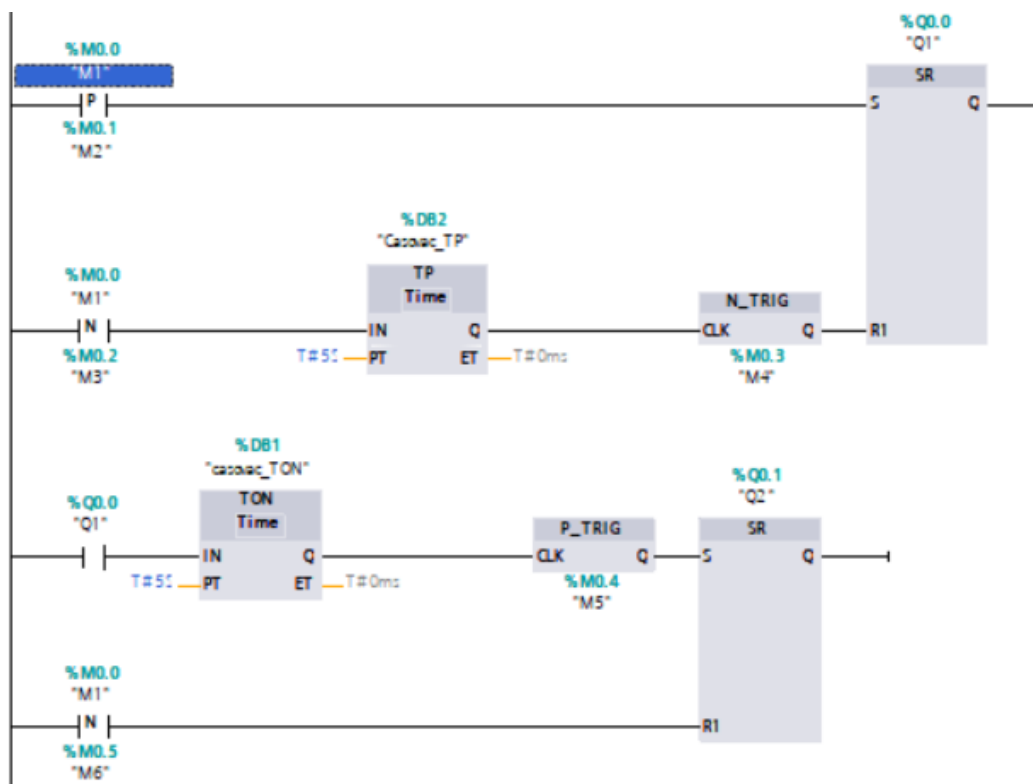
Stav po vypnutí „ $M1$ “ – „casovac\_TOF“ odpočítává čas do vypnutí „ $Q1$ “



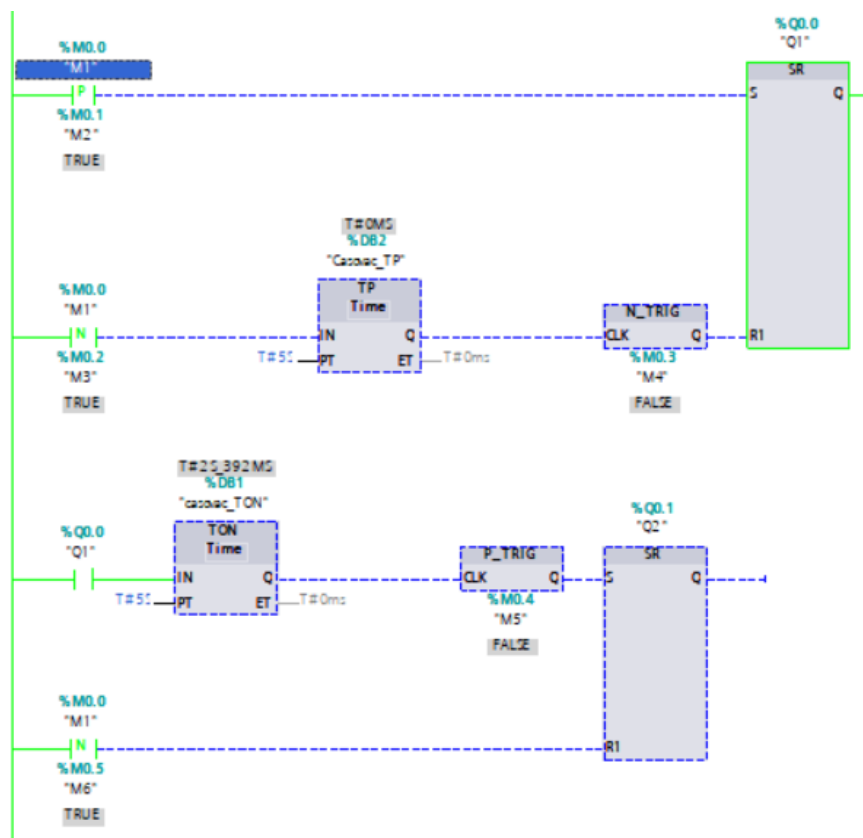
### c) Postupné spínání a vypínání – první se zapíná Q1 a první se vypíná Q2

Zapínání i vypínání je provedeno jedním spínačem „ $M1$ “. Spínač  $M1$  v prvním řádku má pulzní kontakt na vzestupnou hranu. Sepnutím spínače se sepne výstup „ $Q1$ “ bloku „SR“, současně se sepne kontaktem „ $Q1$ “ ve třetím řádku časovač „TON“ (zpožděné zapnutí). Po nastavené době sepne jeho výstup a impulzem bloku P\_TRIG se sepne výstup bloku „SR –  $Q2$ “.

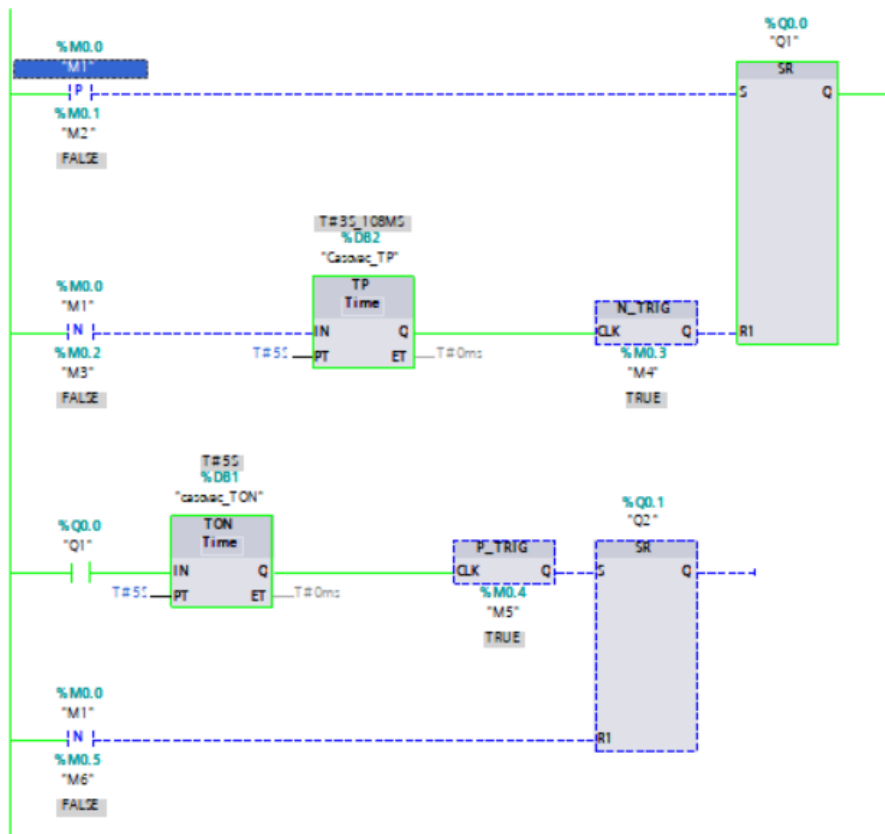
Vypnutím spínače „M1“ dojde k impulzu kontaktu „M1 – N“ ve 4. řádku a 2. řádku. Tím dojde k vypnutí výstupu „SR – Q2“. Současně s vypnutím (M1 – N) je dán impulz časovači „Casovac\_TP“ (zpožděné vypnutí) druhém řádku a začne časovat a po jeho vypnutí výstup blok „N\_TRIG“ dá impulz na R1 (reset) bloku „SR – Q1“ a jeho výstup se vypne .



Stav po sepnutí „M1“, kdy je sepnutý výstup Q1 a časuje zpožděné zapnutí „casovac\_TON“ pro výstup „Q2“.



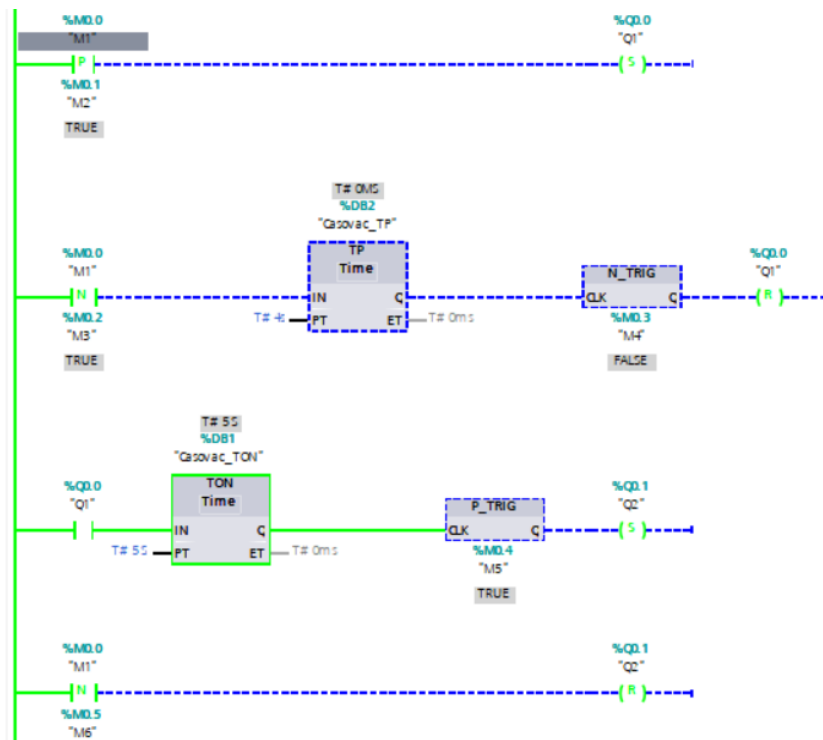
Stav po vypnutí „M1“. Je vypnutý výstup „Q2“ a běží časování „Casovac\_TP“ pro vypnutí „Q1“



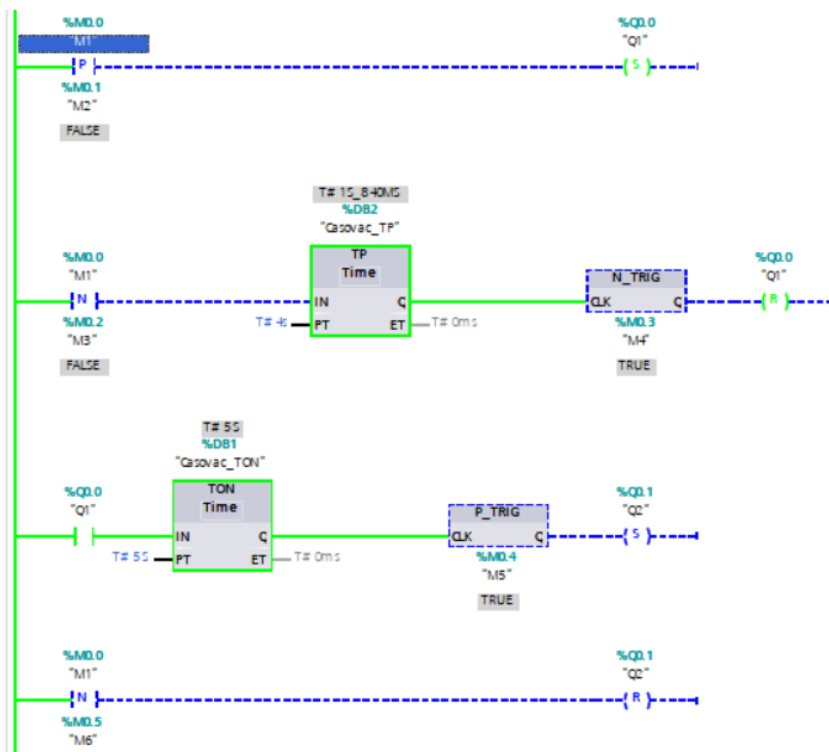
d) Postupné spínání a vypínání – první se zapíná Q1 a první se vypíná Q2

Je to ten samý příklad jako v bodě A), s tím rozdílem, že se nepoužívají bloky SR, ale kontakty S a R výstupů Q.

Stav při zapnutí Q1 i Q2



Stav při vypnutí M1, výstup Q1 je vypnutý a časuje „Casovac\_TP“ – vypnutí „Q1“.

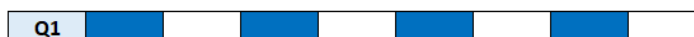


### e) Pulzní generátor

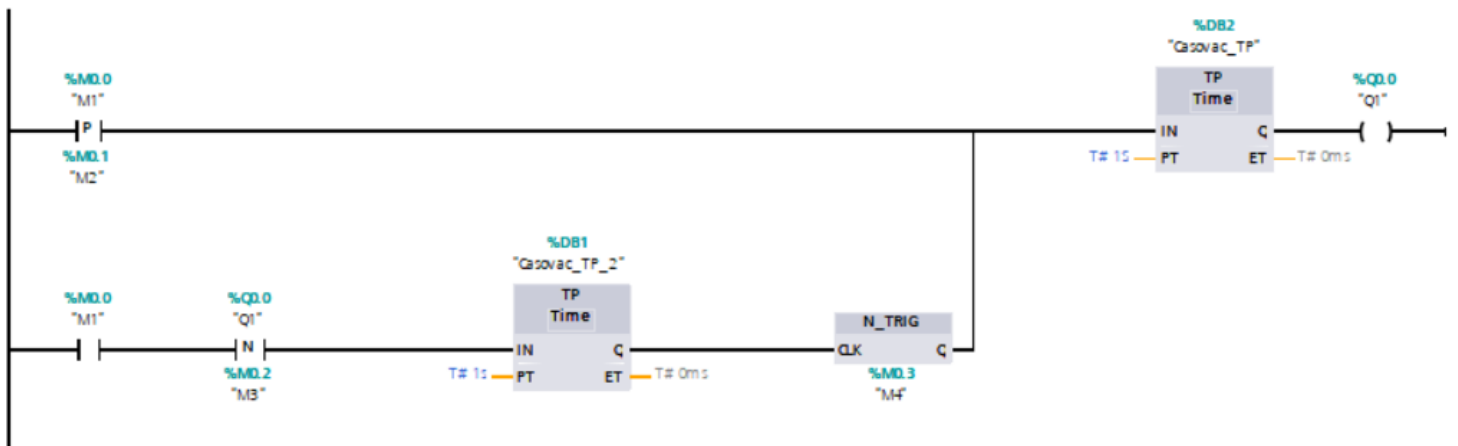
Střídavě se spíná a vypíná výstup Q1

Podle nastavených časů se jedná o synchronní nebo asynchronní generátor

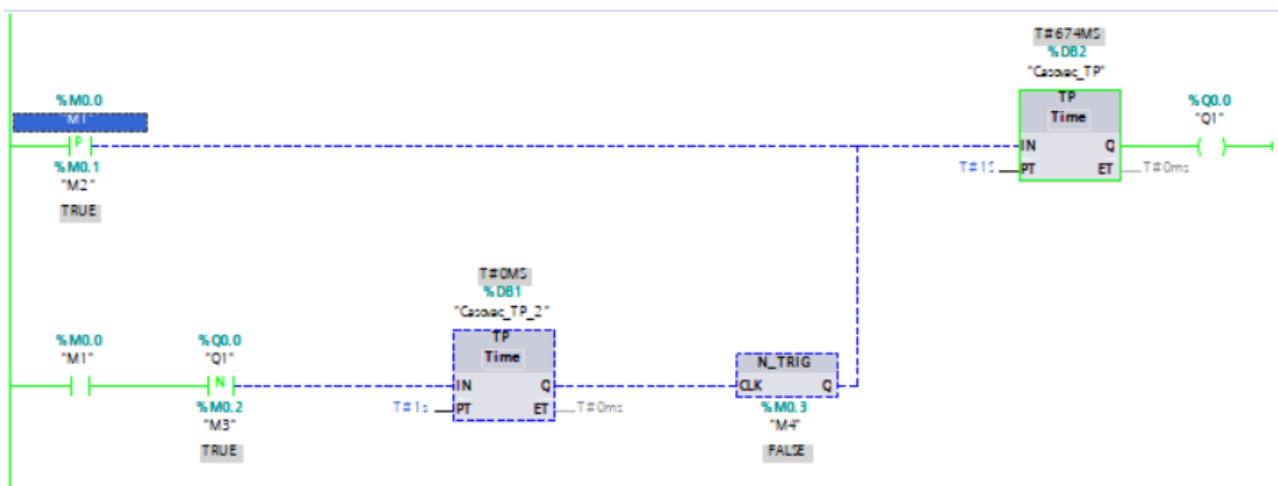
Sepnutím spínače „M1“ se pulzem v 1.řádku sepne časovač „TP“ a výstup „Q1“ ( doba pulzu) po vypnutí se impulzem „Q1 – N“ sepne časovač „Casovac\_TP2“ (doba mezi pulzy). Po jeho vypnutí dá blok „N\_Trag“ impulz na vstup bloku „Casovac\_TP“ a sepne se opět jeho výstup. Tak se až do vypnutí spínače „M1“ střídá zapnutí a vypnutí „Q1“.



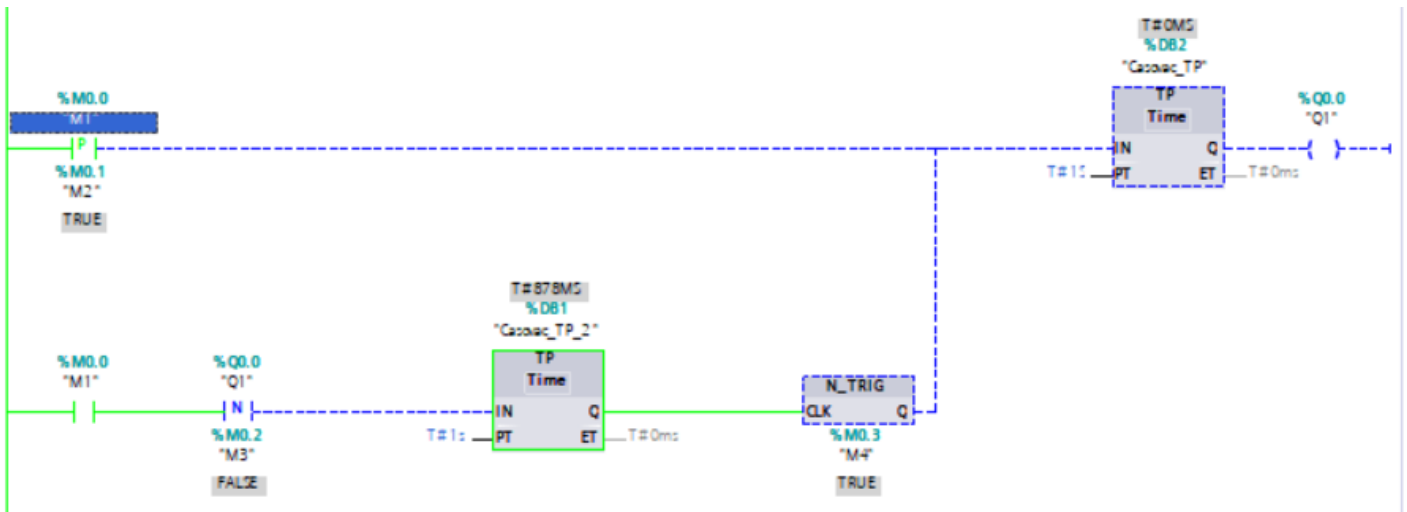
zapnuto	
vypnuto	



Stav při sepnutí „Q1“. Doba pulzu je dána nastavením času PT - „Casovac\_TP“

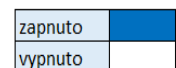


Stav mezi pulzy. Doba mezi pulzy je dána volbou času „Casovac\_TP\_2“.

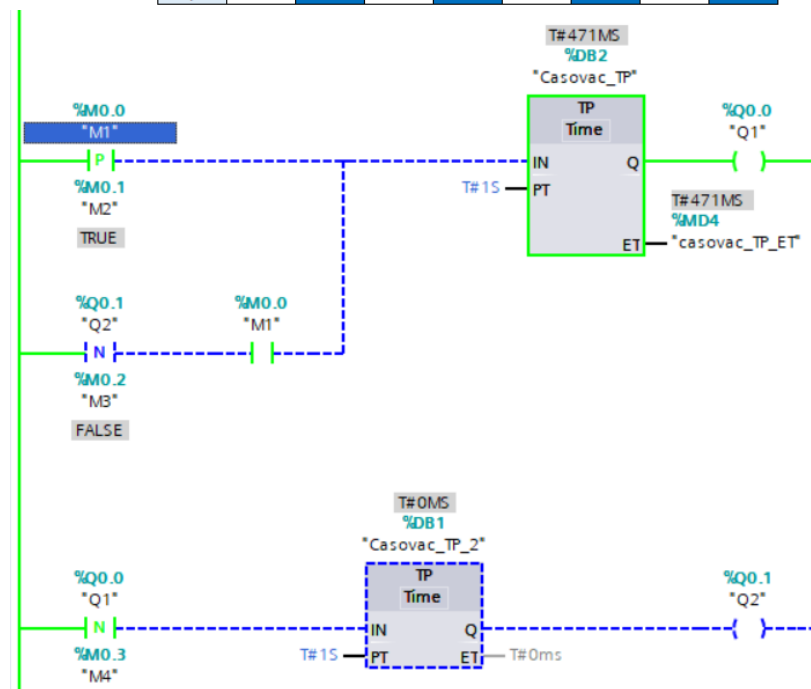


### f) Střídavé spínání a vypínání dvou výstupů

Sepnutím spínače „M1-P“ se dá impulz na vstup IN časovače „Casovac\_TP“. Ten sepne výstup „Q1“ na dobu nastavenou na vstupu PT. Po ukončení časování se vypne výstup Q1, jeho vypnutím dojde k impulzu na 3.řádku „Q1-N“. Tím dojde k sepnutí výstupu časovače „Casovac\_TP\_2“ a „Q2“ na dobu nastavenou na TP časovače.



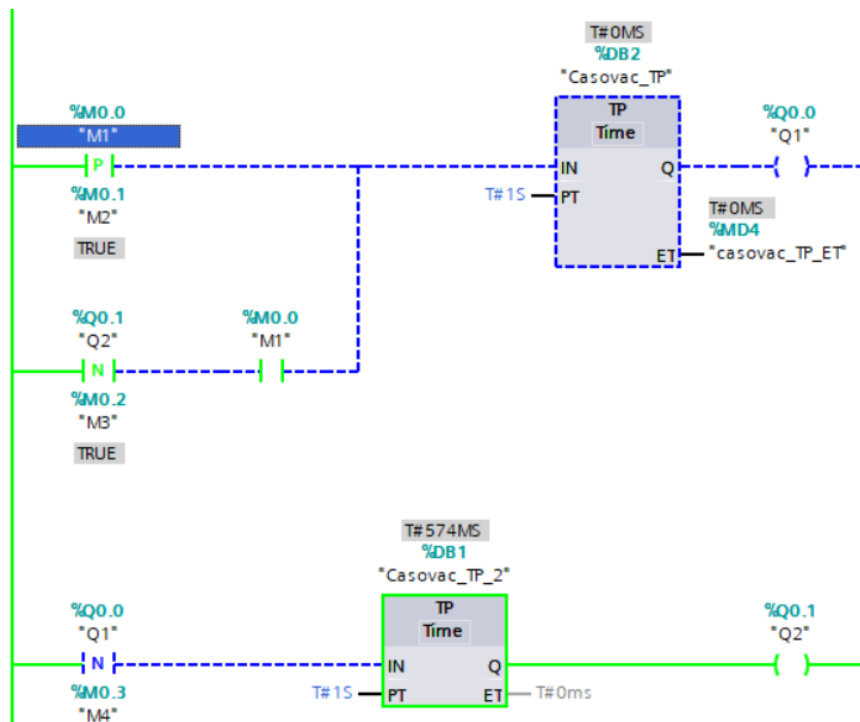
Stav při sepnutí



Stav při sepnutí

Q1

Q2



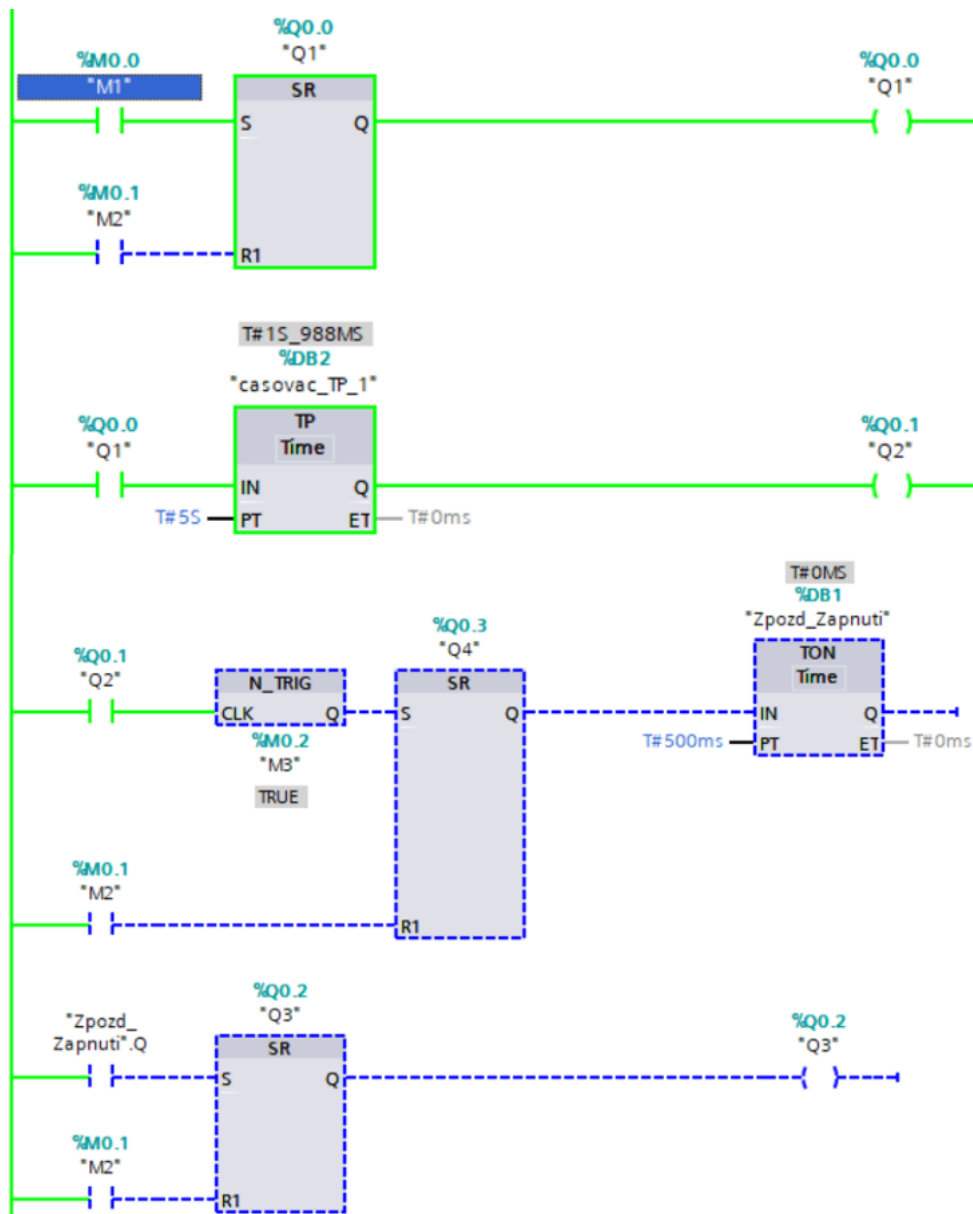
### g) Rozběh motoru hvězda trojúhelník

Spouštění asynchronních elektromotorů nad 4kW je z důvodu snížení proudu při rozběhu prováděno přepínáním jejich zapojení ze zapojení z hvězdy do trojúhelníku. K tomu jsou použity tři stykače. První stykač Q1 přivádí proud na jednu stranu svorkovnice elektromotoru. Stykač Q2 propojuje svorky elektromotoru do hvězdy a třetí stykač Q3 propojí motor do trojúhelníku.

Stiskem tlačítka „M1“ se sepne výstup bloku „SR – Q1“ a výstup „Q1“. Současně se sepne časovač „casovac\_TP\_1“ a výstup Q2 (zapojení do hvězdy). Po ukončení časování se vypne výstup „Q2“ a blok „N\_TRIG“ dá impuls na vstup S bloku „SR – Q4“ a sepne jeho výstup Q. Tím začne časovat blok „Zpozdz\_Zapnut“ (prodleva mezi odpojením stykače do hvězdy a sepnutím stykače do trojúhelníku). Po sepnutí jeho výstupu se sepne blok „SR – Q3“ a výstup „Q3“, zapojení do trojúhelníku.

Vypnutí se provede stiskem tlačítka „M2“. Tím se resetují všechny bloky SR.

### Stav při zapnutí do hvězdy

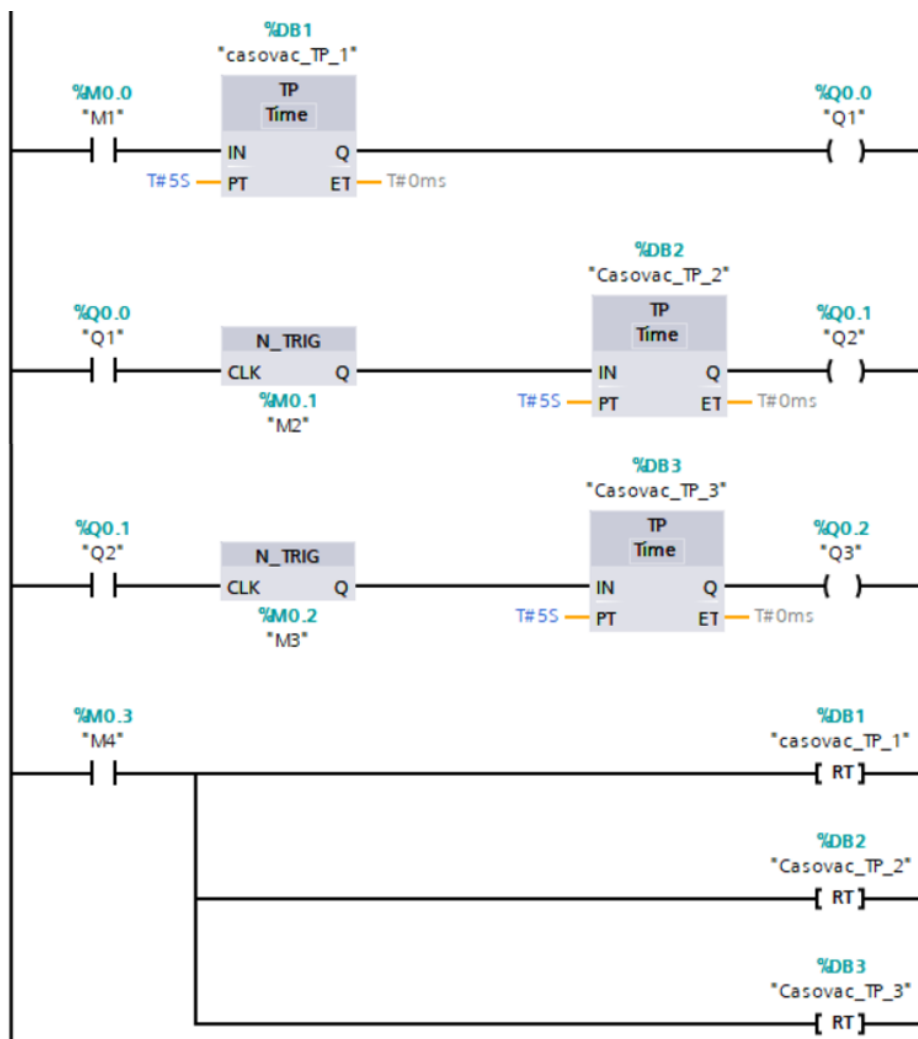


### h) Krokový posun

Po stisku tlačítka „M1“ se sepne výstup „Q1“, po nastaveném čase t1 se vypne a sepne se výstup „Q2“. Ten se po nastaveném čase t2 vypne a sepne se výstup „Q3“, ten se po nastaveném čase t3 vypne. Vypnutí během časování je zajištěno bloky časovačů RT. Blok „N\_TRIG“ dává impuls při vypnutí Q a tím se sepne časovač a jeho výstup.

	t1	t2	t3
Q1			
Q2			
Q3			

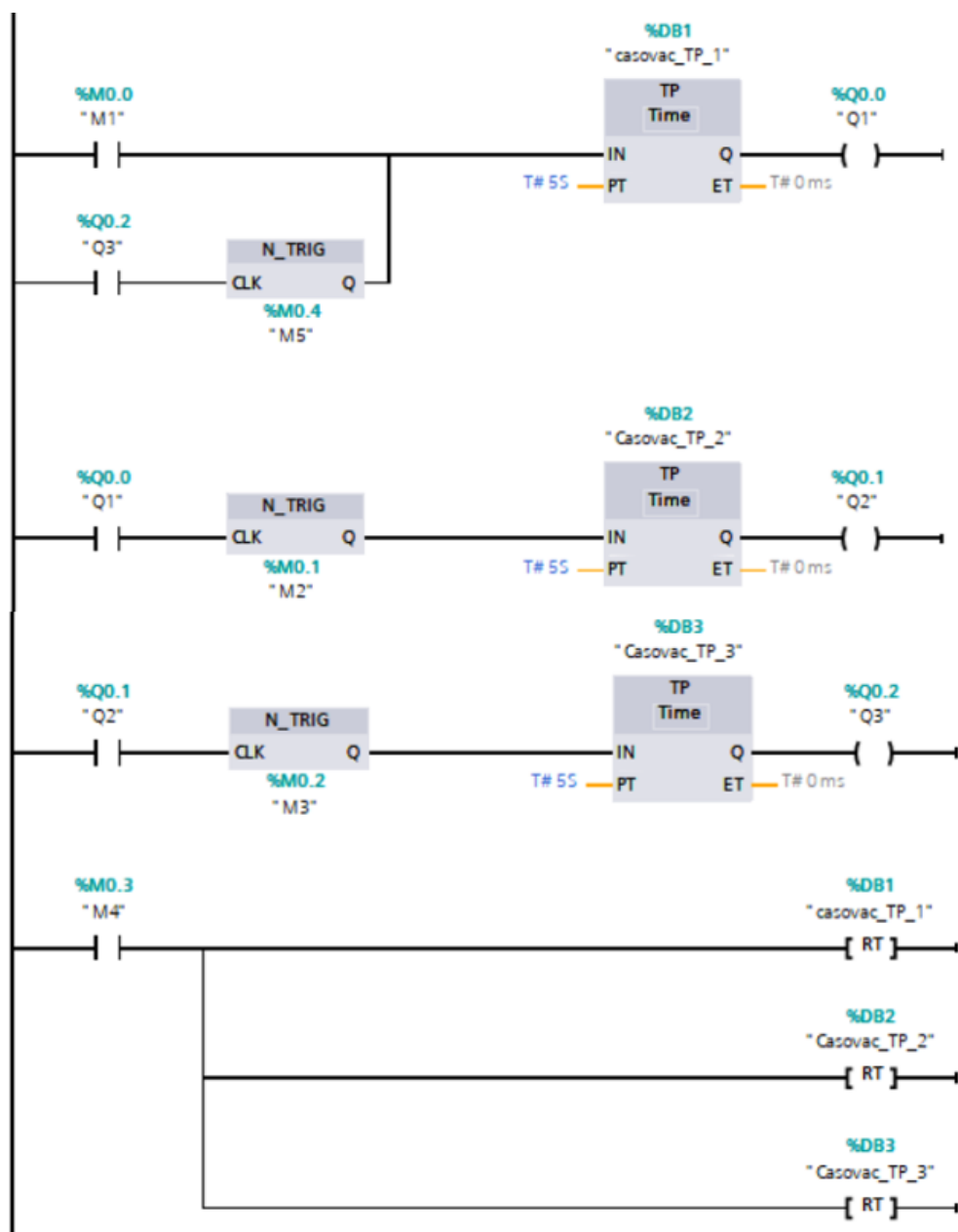
zapnuto	
vypnuto	



### ch) Cyklické spínání výstupů

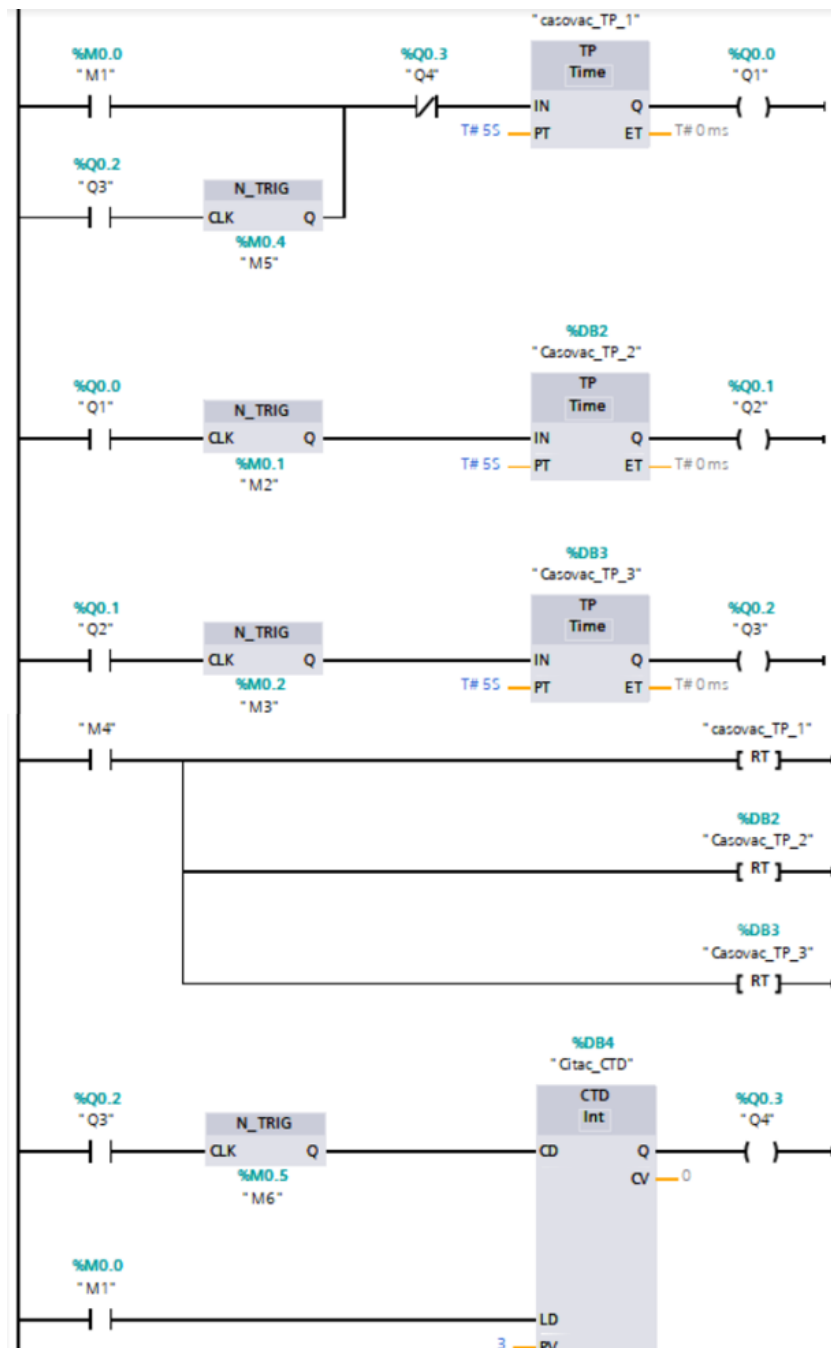
Zde je příklad stejný, jako je ten předchozí „Krokový posun“, ale s tím rozdílem, že vypnutím výstupu Q3 se opět sepne výstup „Q1“ a postup se opakuje.

Do programu je do druhého řádku vložen spínací kontakt „Q3“ a blok „N\_TRIG“, ten dá impulz po vypnutí „Q3“ na vstup IN časovače „casovac\_TP\_1“ a ten sepne výstup „Q1“



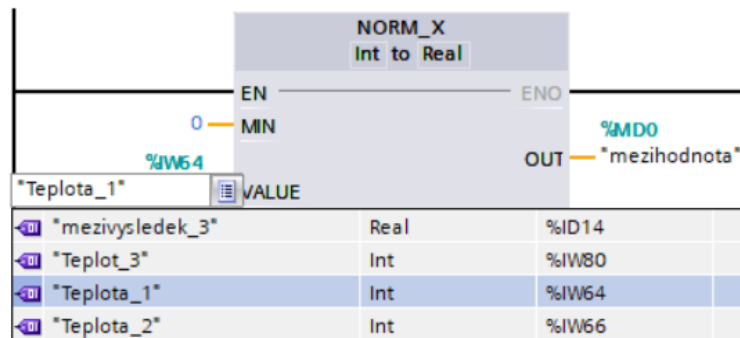
i) Cyklické spínání výstupů s konečným počtem cyklů

V tomto případě se jedná o rozšířený předchozí příklad. Zde je nastaveno provedení 3 cyklů a po jejich proběhnutí se cyklus zastaví. V prvním řádku je přidán rozpínací kontakt výstupu „Q4“. Na konci programu je přidán spínací kontakt „Q3“ na něj je napojen blok „N\_Trig“ a čítač CTD (odečítá od nastavené hodnoty a při dosažení hodnoty 0 se sepne jeho výstup). Při vypnutí „Q3“ dá blok „N\_TRIG“ impuls na vstup CD čítače a jeho hodnota se sníží o 1. Při dosažení hodnoty 0 se sepne výstup čítače Q a výstup Q4. Tím se vypne rozpínací kontakt „Q4“ v prvním řádku a cyklus se zataví. Opětovné zpuštění se provede tlačítkem „M1“, jeho stiskem se nastaví i počáteční hodnota čítače (poslední řádek „M1“ dá impuls na vstup LD čítače, tím se nastaví na hodnotu 3).



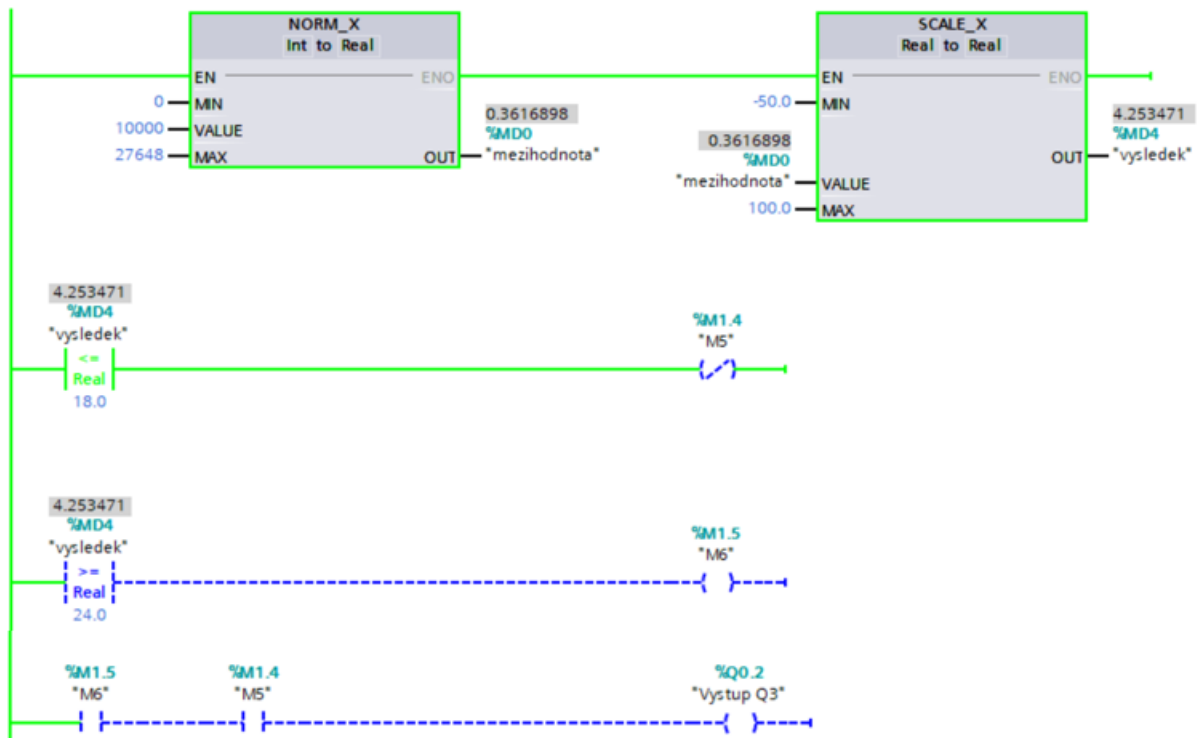
## j) Větrání nebo chlazení

Zde je příklad, kdy se spínají ventilátory při dosažení teploty 24°C a vypínají při poklesu teploty 18°C. Z důvodu simulace je na vstupu „VALUE“ bloku „NORM\_X“ číselná hodnota. V praktickém příkladu zde bude adresa analogového vstupu např. „Teplota\_1“ s adresou IW64.



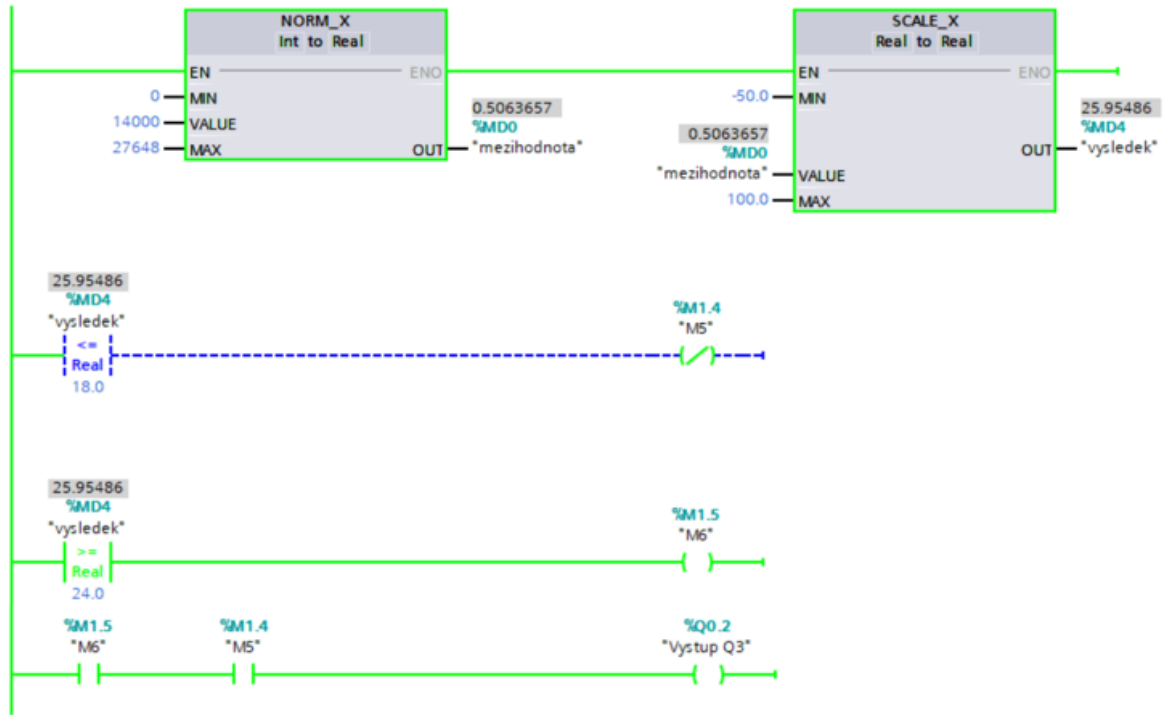
Hodnoty z převodníku se v bloku „Norm\_X“ převedou na hodnoty v intervalu od 0 do 1. V bloku „SCALE\_X „ se v závislosti na nastaveném rozsahu teplot (zde -50 až 100°C), rozsah teplot musí odpovídat převodníku, se vypočte skutečná teplota.

Na obrázku je simulace programu, kdy naměřená teplota 10°C je nižší než teplota nastavená na komparátoru  $\leq$  ve druhém řádku. Ta je 18°C. Teplota pro nastavená pro sepnutí ventilátoru je 24°C (třetí řádek). Výstup pro spínání ventilátorů je „Q3“. Je-li teplota nižší než 18°C je výstup z komparátoru na 2. řádku sepnutý. Vypnutý je druhý komparátor na 3. řádku, protože teplota není vyšší než 24°C. Ve druhém řádku je negovaný výstup M5, je tedy vypnutý. Ve 4. řádku jsou za sebou oba spínací kontakty „Vystup M5“ a Vystup M6“ vypnuté. Proto je výstup „Q3“ vypnutý.



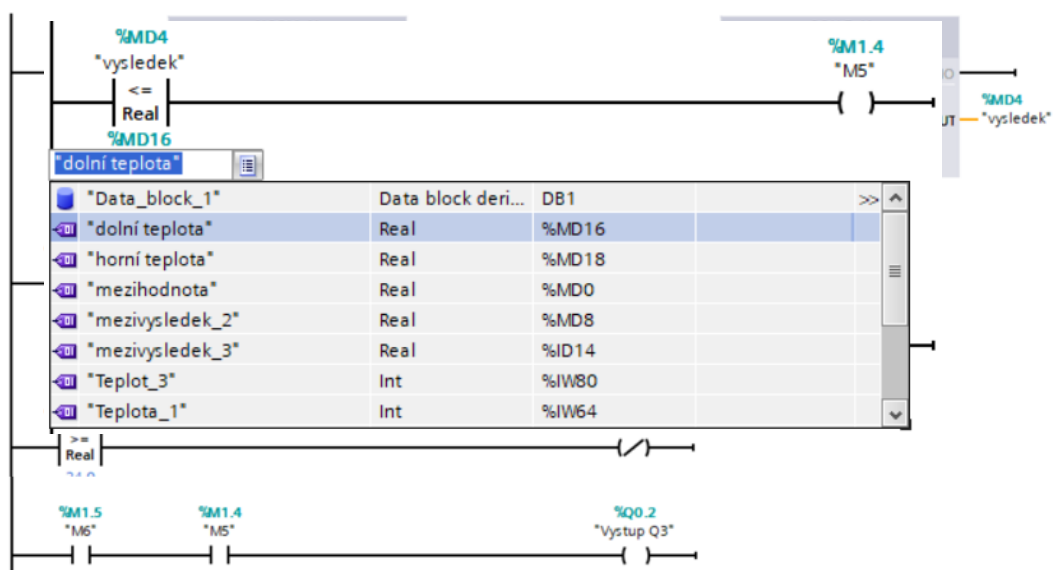
Na tomto obrázku je měřená teplota 25,9°C. Nastavená teplota, kdy dojde k sepnutí ventilátorů je 24°C. Protože je teplota vyšší než 18°C je komparátor  $\leq$  ve druhém řádku vypnutý, když je vypnutý je sepnutý rozpínací kontakt „M5“. Teplota 25,9 °C je vyšší než hodnota na komparátoru  $\geq$  ve 3.řádku

24°C proto je spínací kontakt „M6“ sepnutý. Ve 4.řádku jsou kontakty M6 a M5 sepnuté proto je sepnutý výstup „Q3“ a jsou spuštěné ventilátory.

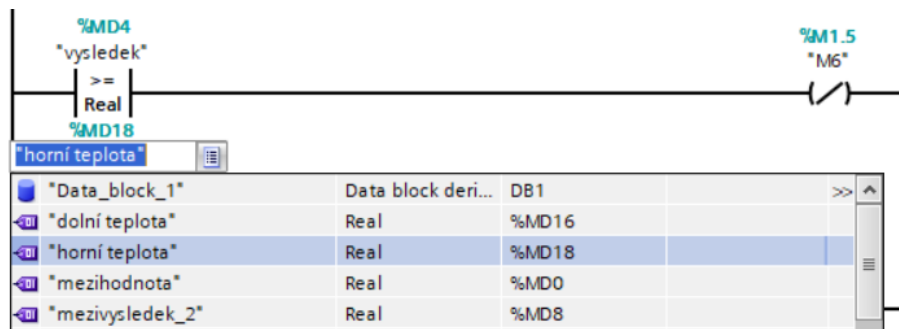


### k) Topení

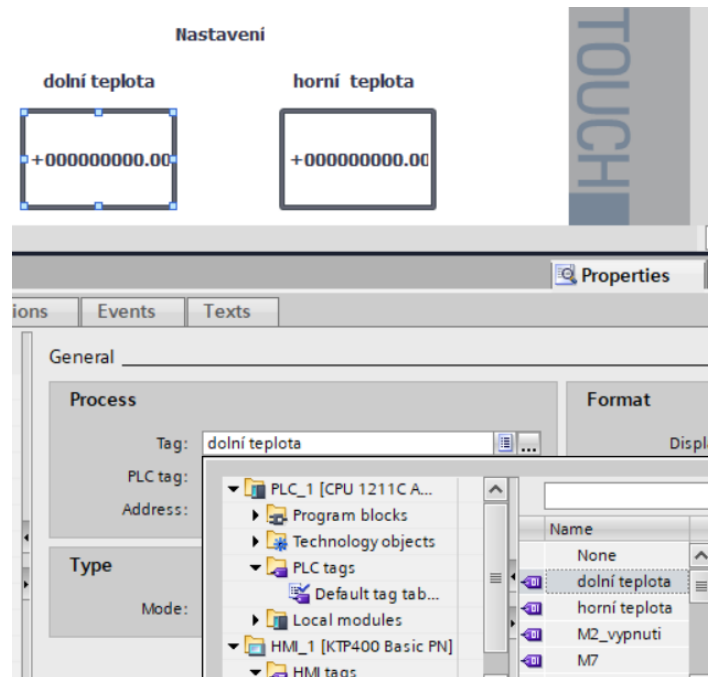
Je-li teplota v tomto příkladu nižší nebo rovna 18°C, sepne se vytápění, při 24°C se vypne. Zapojení je podobné jako v předchozím příkladu. Rozdíl je v tom, že výstup „M5“ je spínací a „M6“ rozpínací. Pokud budeme chtít měnit nastavení teplot např. na panelu HMI vložíme ve spodní části komparátoru místo pevné teploty adresu. Vložení adresy do bloku ≤.



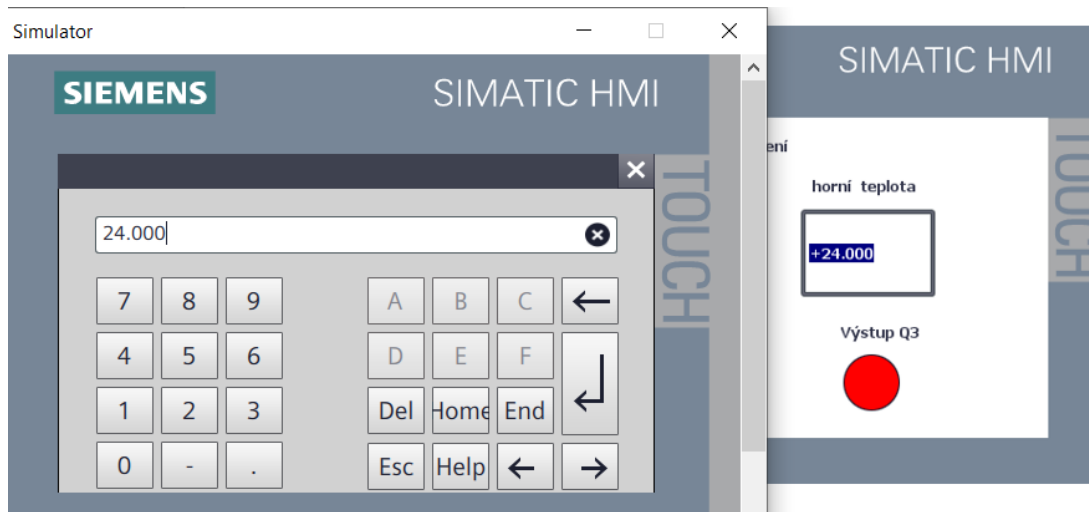
Vložení adresy do bloku ≥ .



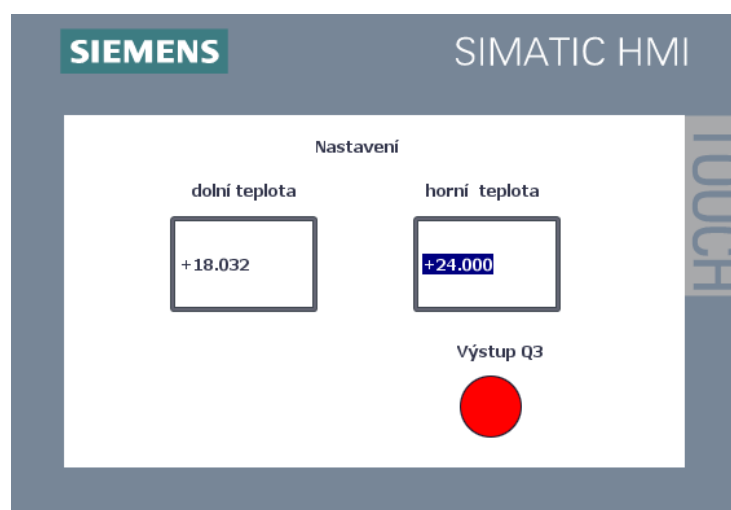
Na panel HMI vložíme blok „I/O field“ a vyhledáme adresu v PLC tags, kterou jsme vložili do komparátoru.



Kliknutím do bloku zobrazovače čísel na panelu HMI nastavíme na klávesnici, které se zobrazí, požadovanou teplotu.

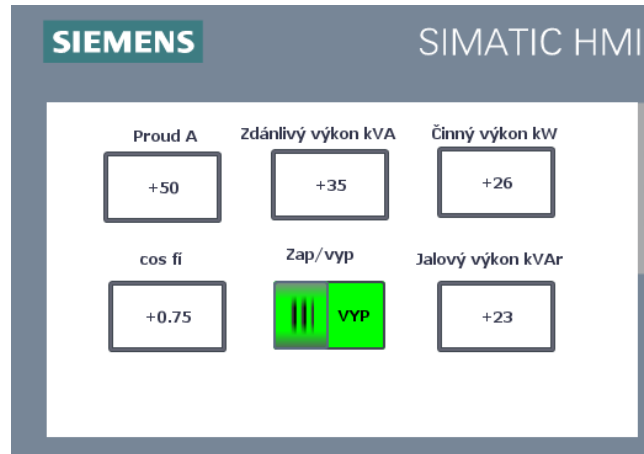


Nastavené teploty



## I) Měření spouštění a jištění asynchronního motoru

Tento příklad kombinuje analogové bloky, matematické bloky a komparátor. V příkladu se panelu na MHI zobrazuje celkový proud, činný výkon, zdánlivý výkon, jalový výkon a  $\cos \varphi$ , kde  $\varphi$  je fázový posuv mezi fázovým napětím a fázovým proudem a je zde spínač pro sepnutí motoru. Při přetížení nad nastavenou hodnotu se motor vypne.



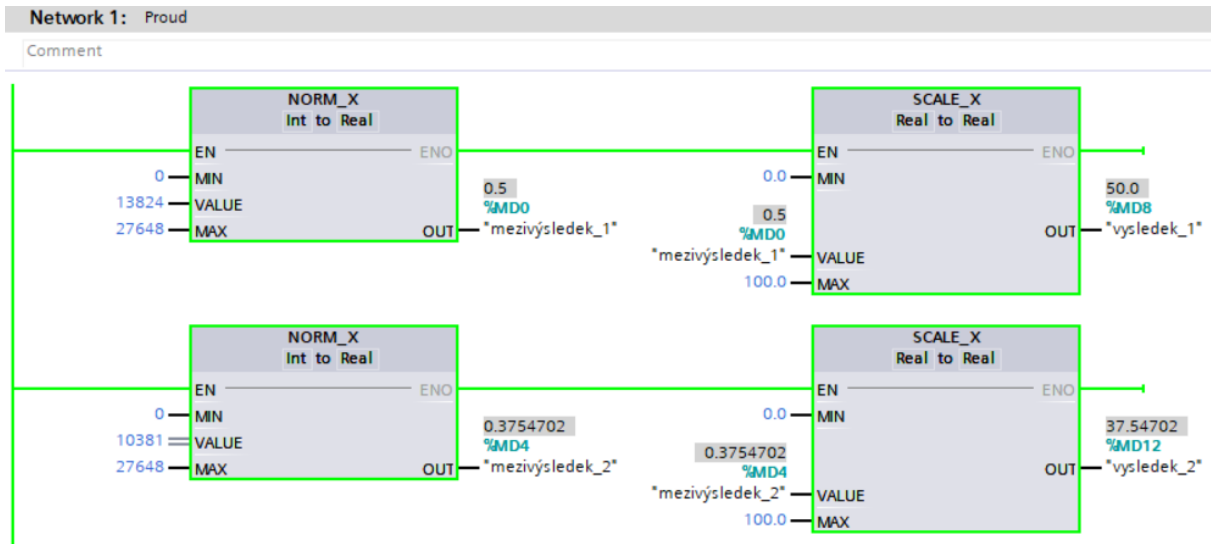
1. zapíšeme názvy adresy analogových vstupů Properties → IO tags (proud)

The screenshot shows the SIMATIC Manager software interface. On the left is the project tree, and on the right is the 'Device overview' window. The 'IO tags' tab is active, showing the following table:

Name	Type	Address	Tag table
Proud I	Int	%IW64	Default tag table
Proud Q	Int	%IW66	Default tag table

Program je pro přehlednost zapsán v samostatných oknech Network. Ke každému Network: můžeme zapsat název (Proud)

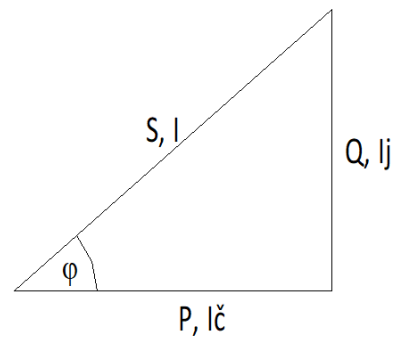
V této části je v prvním řádku zpracován celkový proud z převodníku proud/napětí. Ve druhém řádku je jalový proud. Výstup proudu z prvního řádku má adresu „výsledek\_1“ a v uvedeném příkladu je jeho hodnota „50“. Ve druhém řádku je výstup a má adresu „výsledek\_2“ a má hodnotu 37,547.



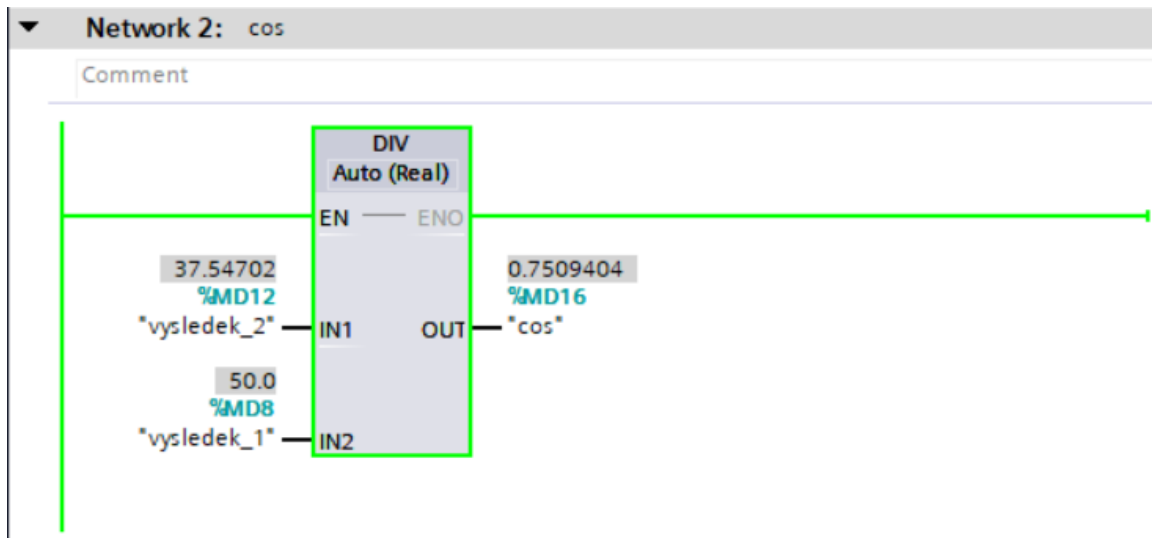
Výpočet  $\cos \varphi$

$$\cos \varphi = I_j / I$$

Na obrázku jsou znázorněny proudy a výkony. Výkon je úměrný proudu.



Pro výpočet  $\cos \varphi$  použijeme blok dělení. Dělíme „výsledek\_2“ vstupem IN1 vstupem IN2 „výsledek\_1“.  
 ( $\cos \varphi = 37,547/50 = 0,75$ )



Výpočet zdánlivého výkonu S

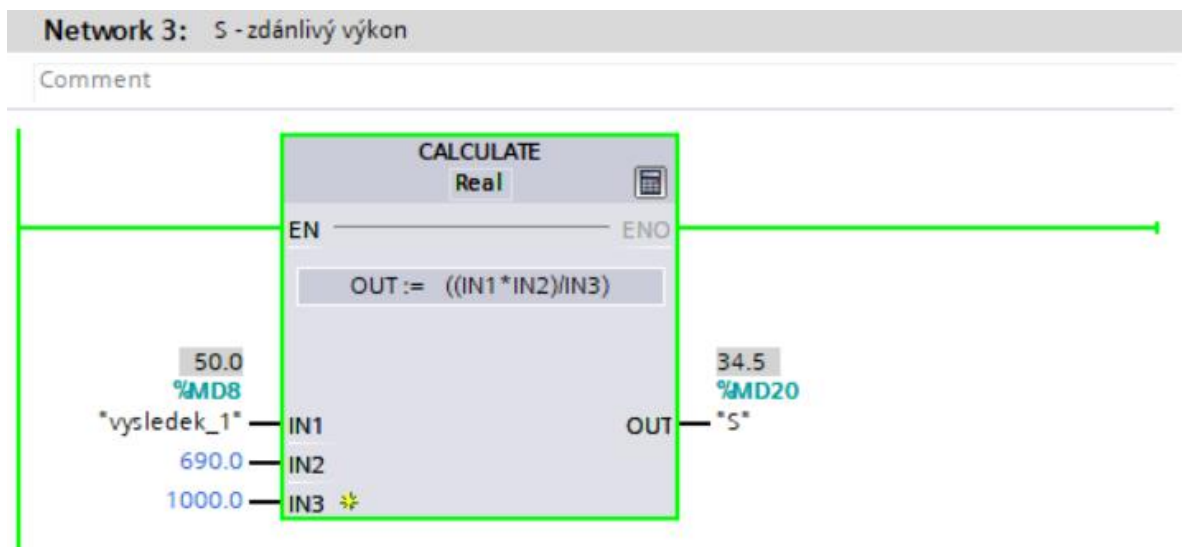
Pro výpočet použijeme matematický blok „Calculate“ (spočítat)

$$S = 3 \cdot U_f \cdot I / 1000 = 3 \cdot 230 \cdot 50 / 1000 = 34,5 \text{ kVA}$$

Na vstupu IN1 je celkový proud.

Na vstupu IN2 je násobek  $3 \times 230\text{V} = 690\text{V}$

Na vstupu IN3 je 1000, to je převod VA na kVA



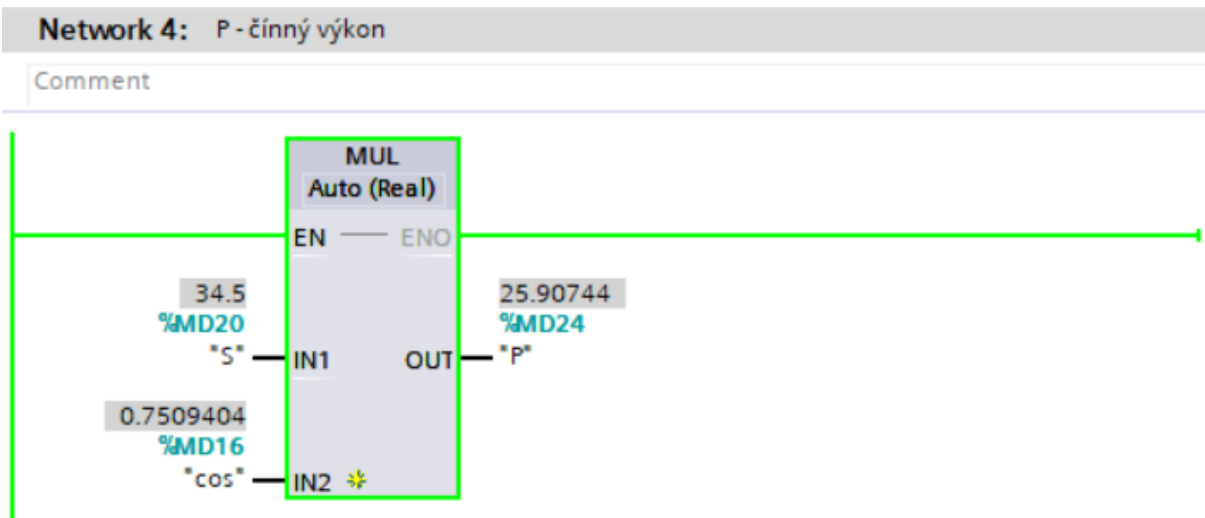
## Výpočet činného výkonu

Pro výpočet činného výkonu byl použitý blok pro násobení „MUL“ .

$$\cos \varphi = P/S \Rightarrow P = S \cdot \cos \varphi = 34,5 \cdot 0,75 = 25,9 \text{ kW}$$

Na vstupu IN1 je použitý výstup „S“ z části programu „Network 3“

Na vstupu IN2 je použitý výstup „cos“ z části programu „Network 2“

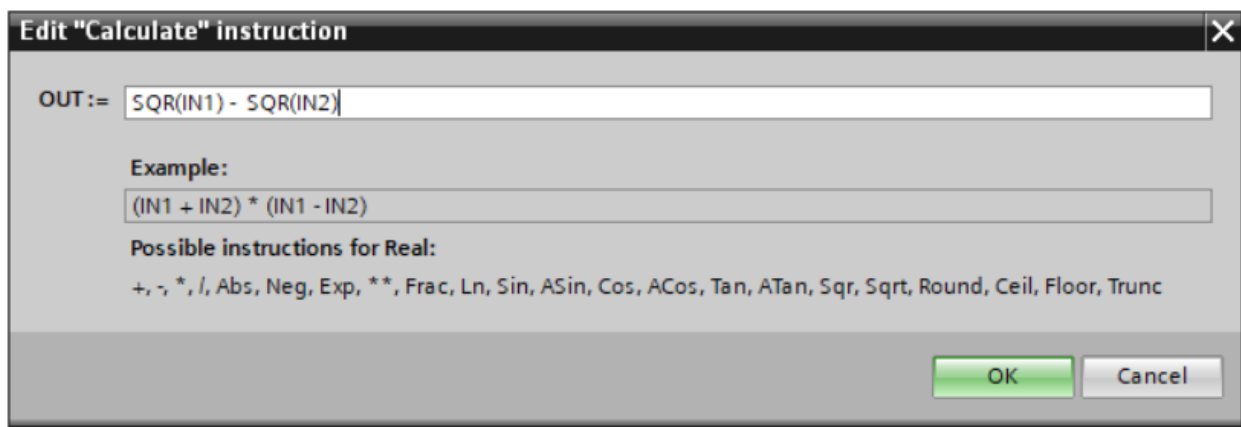


## Výpočet jalového výkonu Q

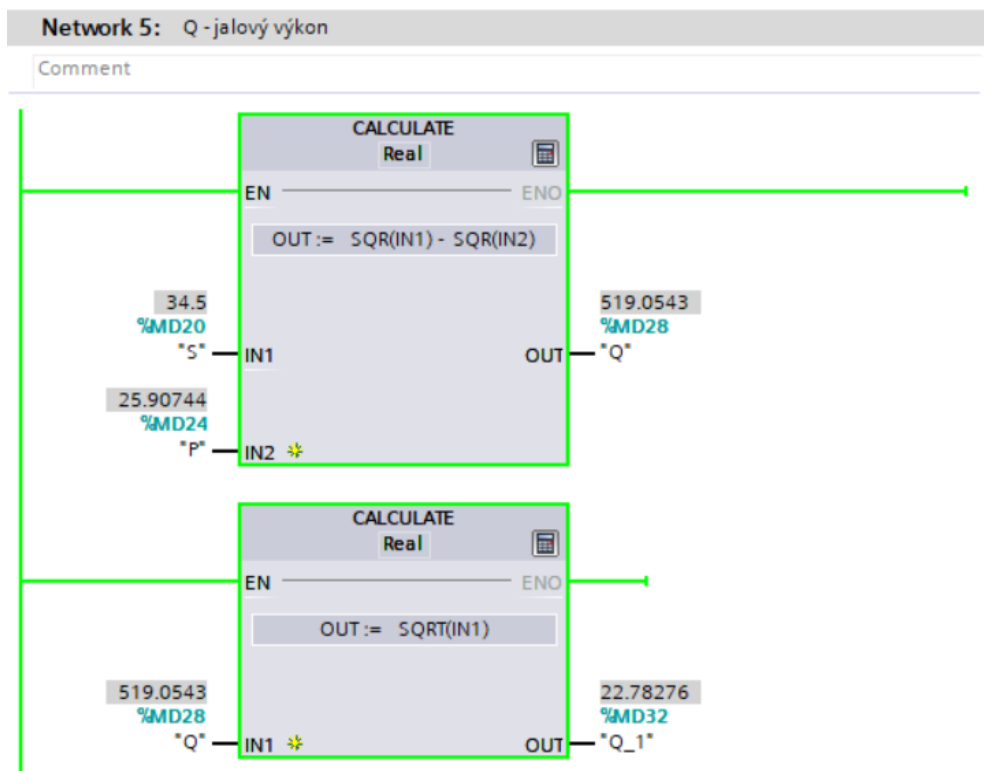
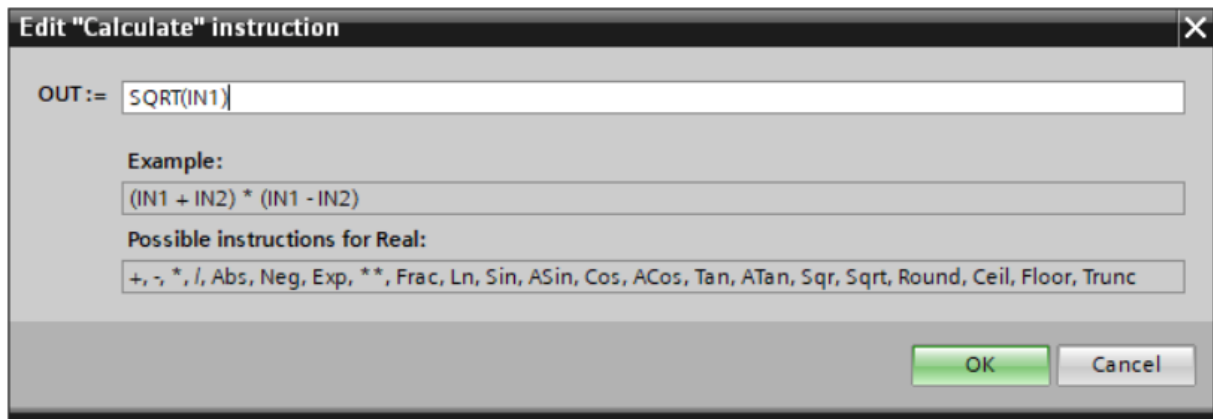
Pro výpočet jalového výkonu byla použita Pythagorova věta, z důvodu ukázky výpočtu mocnin a odmocniny . V příkladu jsou použity dva bloky „Calculate“

$$Q = \sqrt{S^2 - P^2} = \sqrt{35^2 - 26^2} = 23 \text{ kVAr}$$

Blok „Calculate“ v prvním řádku počítá rozdíl dvou mocnin.  $IN1^2 - IN2^2$  .

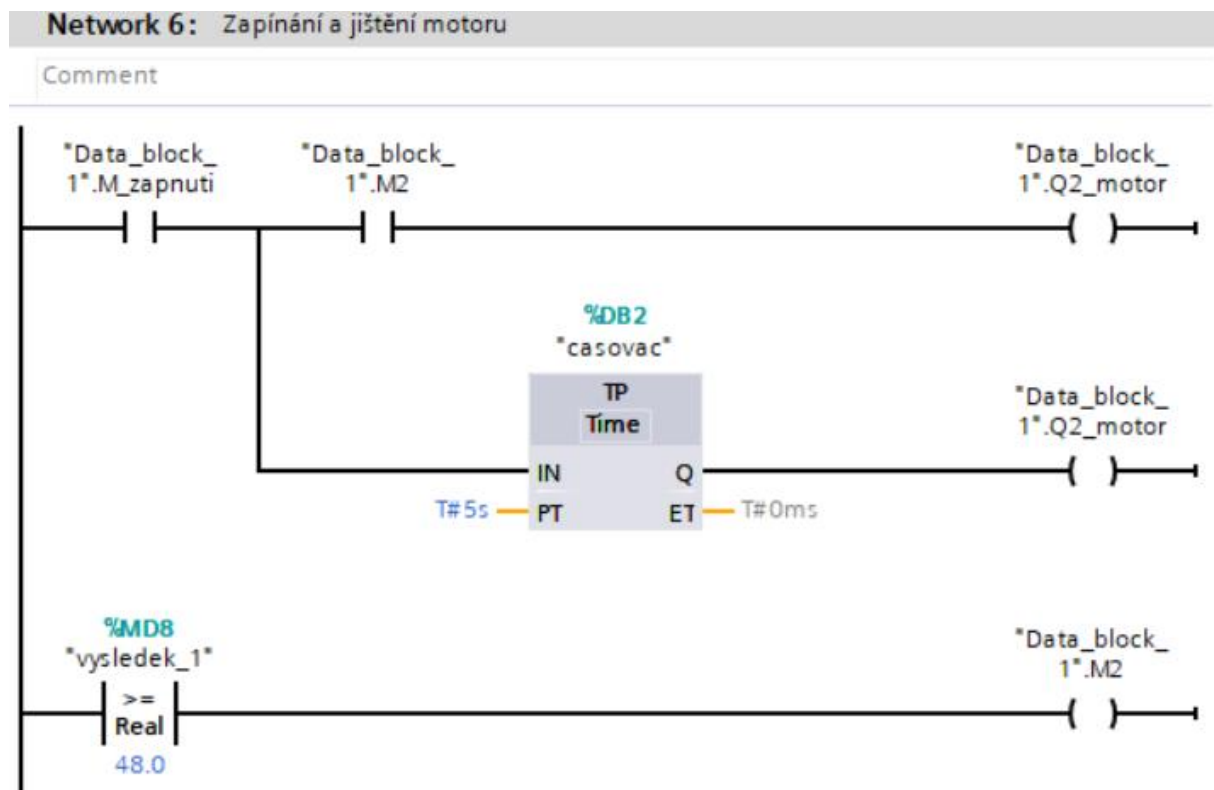


Blok „Calculate“ ve druhém řádku počítá odmocninu výstupu z prvního řádku „Q“.

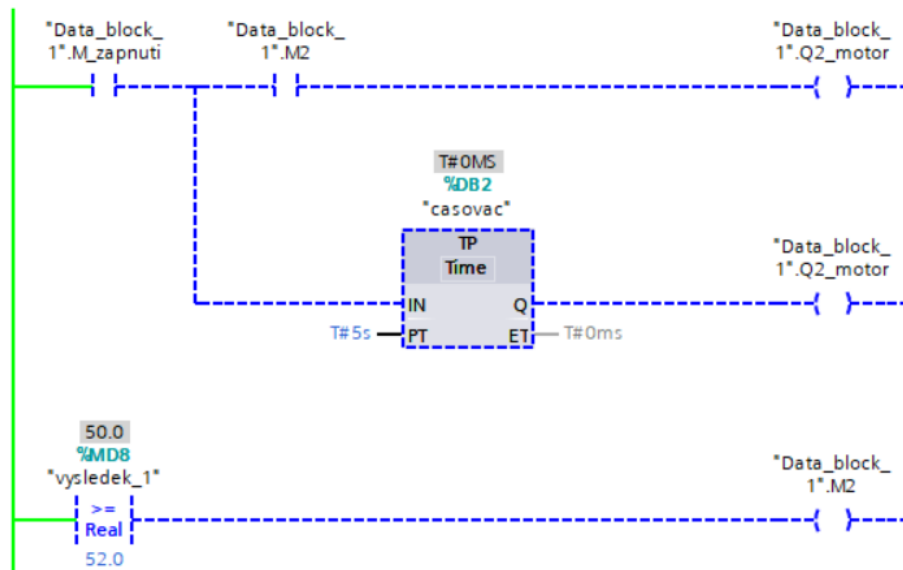


Poslední část programu řeší zapnutí motoru a jeho vypnutí při zvýšeném zatížení než je nastavené.

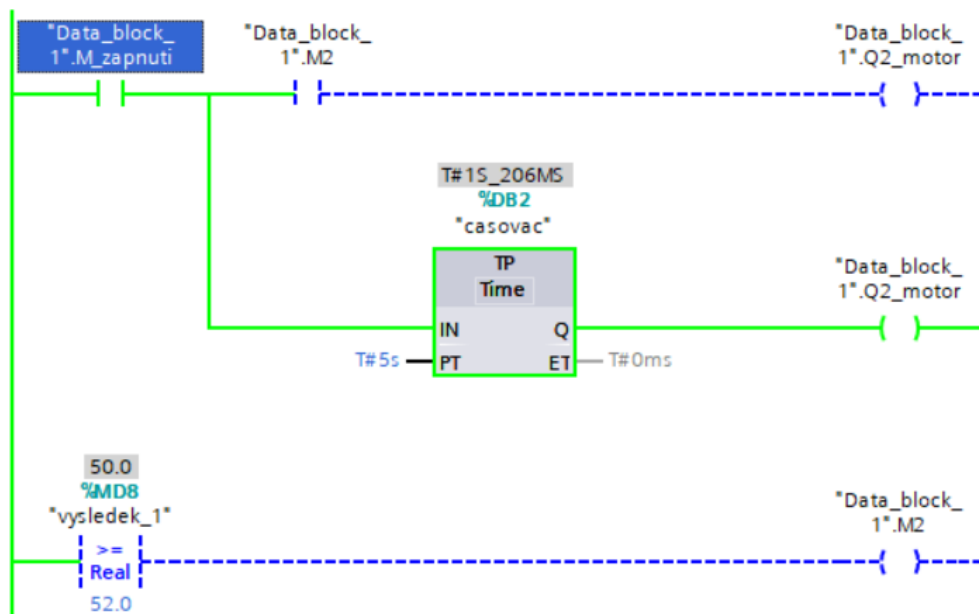
V prvním řádku spínací kontakt s adresou „Data\_block\_1.M\_zapnuti“ zapíná motor. Zapnutí se provede na panelu HMI. Druhý kontakt v sérii je „Data\_block\_1.M2“, ten navazuje na třetí řádek, kde je komparátor  $\geq$ . Je-li proud nižší než 48 A výstup se sepne druhý spínací kontakt v prvním řádku a tím i výstup Data\_block\_1.Q2\_motor. Při spouštění motoru dochází k několikanásobnému nárustu proudu (5x až 7x). Tím by došlo k okamžitému vypnutí kontaktu „Data\_block\_1.M2“ a motor by se nespustil, proto je do druhého řádku vložen časovač „TP“. Ten po dobu rozběhu sepne „Data\_block\_1.Q2\_motor“. Není-li motor přetížen sepne se „Data\_block\_1.M2“ a motor běží trvale.



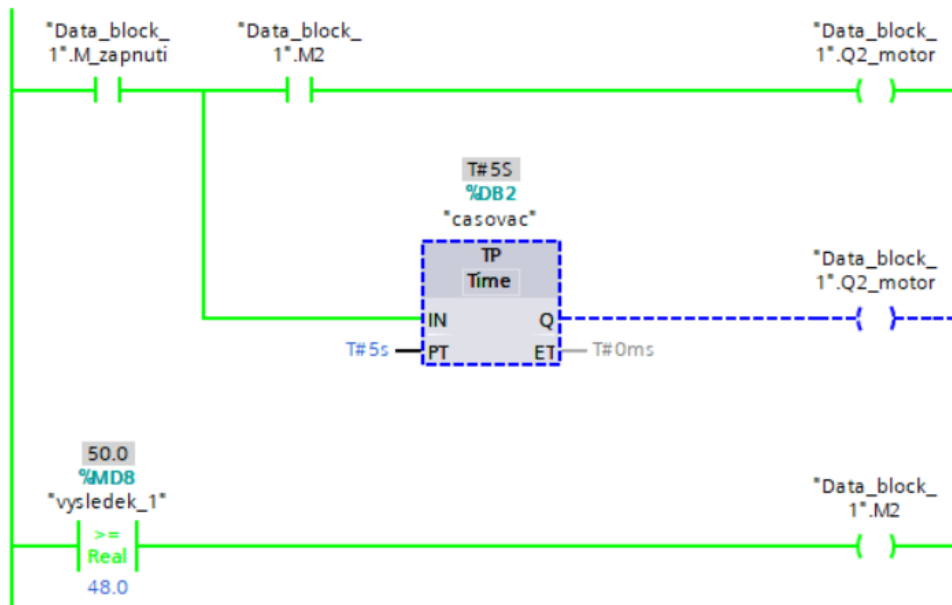
### Stav před spuštěním



Stav po sepnutí, chod motoru zajišťuje sepnutý časovač.



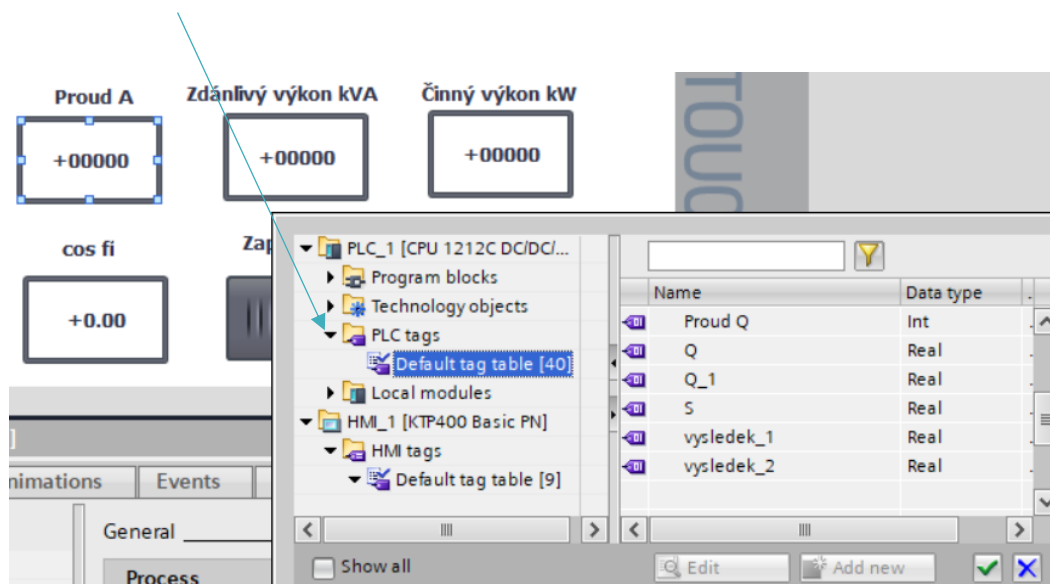
Stav po rozběhu motoru – trvalý chod.



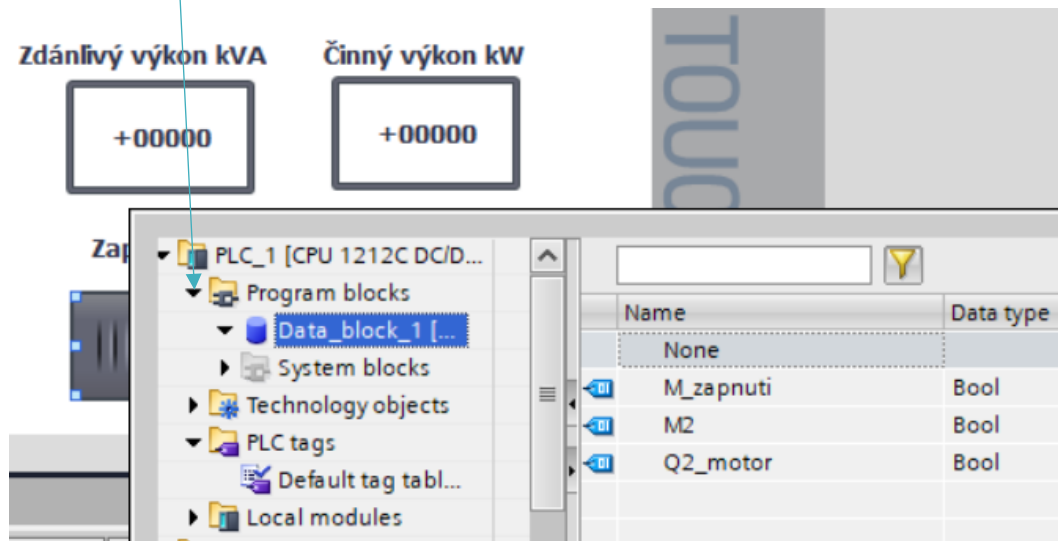
Zobrazení na panelu HMI

Adresy pro bloky musíme hledat tam kam jsme je uložili:

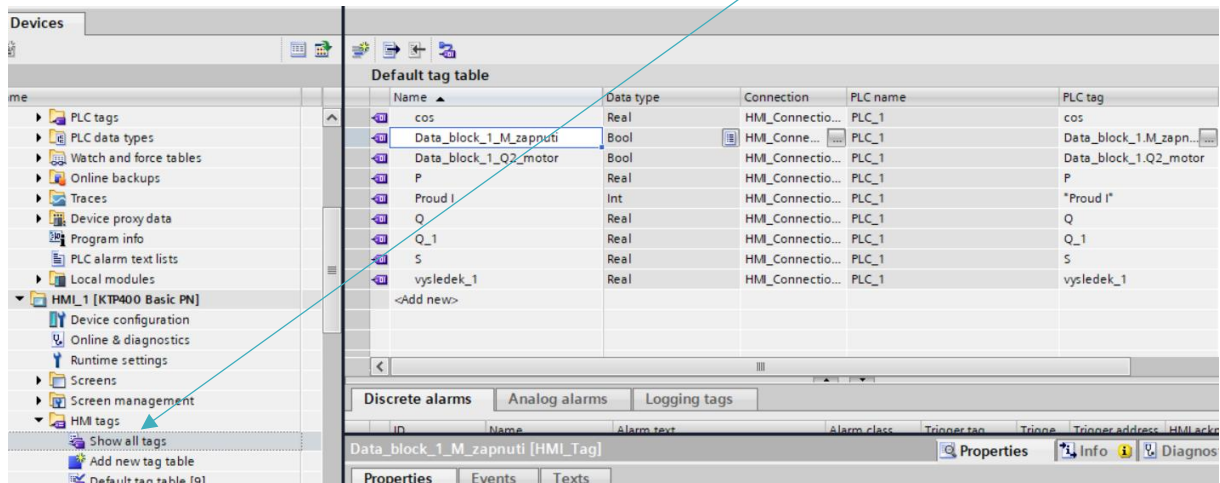
Proud A: PLC tags → Default tag table → výsledek\_1



Zap/vyp: Program blocks → Data\_block\_1 → M\_zapnuti



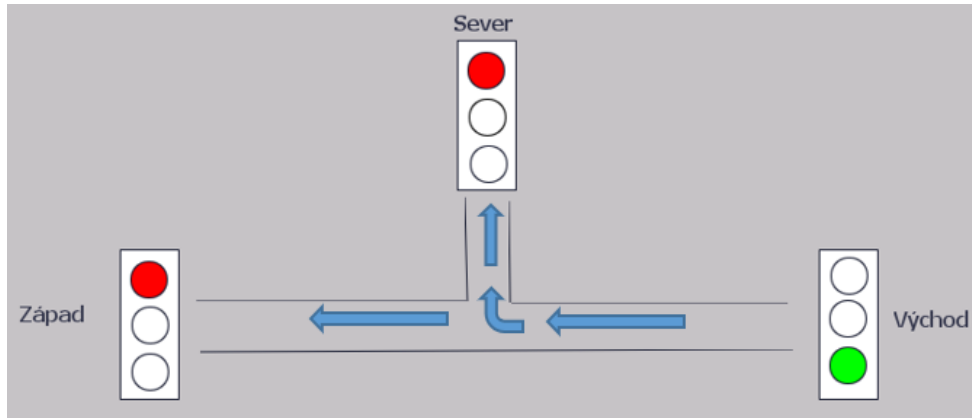
Adresy, které vybereme se zobrazí v HMI tags. Pokud je budeme používat opakovaně např, jednou pro zobrazení proudu číselně a podruhé v grafu, můžeme adresu grafu zadat z HMI tags.  
k



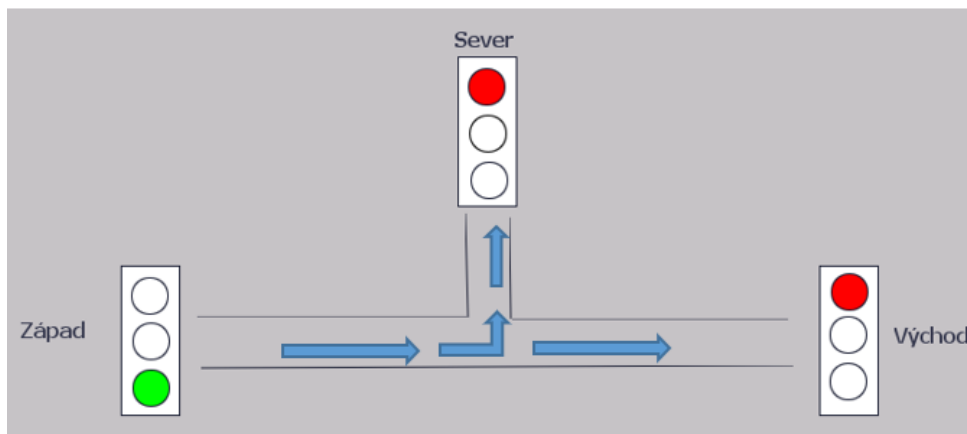
### m) Semafor

Třemi semaforu je řízena křižovatka tvaru T. Na následujících obrázcích jsou znázorněny tři stavy průjezdu křižovatkou.

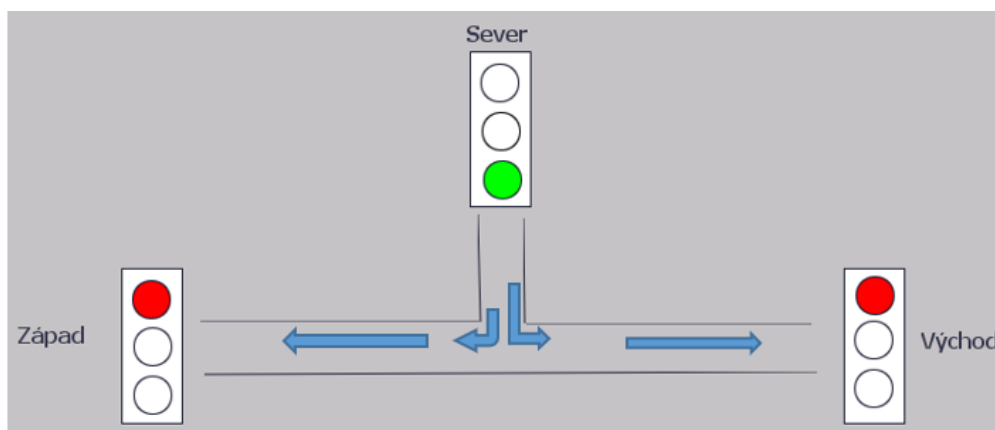
obr.1



obr.2



obr.3



Rozsvěcení semaforů zapíšeme do tabulky, ze které budeme vycházet při programování rozsvěcení jednotlivých semaforů.

obr.4

č.ř	semafor sever	semafor západ	semafor východ	krok
1.	červená	červená	zelená	krok1
2.	červená	červená	žlutá	
3.	červená	zelená	červená	krok2
4.	červená	žlutá	červená	
5.	zelená	červená	červená	krok3
6.	žlutá	červená	červená	

Protože jsou semafony tři rozdělíme postup spínání do třech kroků. Za výchozí vezmeme semafor východní. Bude-li svítit po spuštění zelená, bude na západním a severním semaforu svítit červená (řádek 1.). Po rozsvícení žluté na východním semaforu bude na západním a severním semaforu svítit stále červená (řádek 2.). Po zhasnutí žluté na východním semaforu se rozsvítí na západním semaforu zelená a na ostatních dvou červená (řádek 3.). Po rozsvícení žluté zůstane svítit na semaforu sever a východ červená (řádek 4.). Po zhasnutí žluté na semaforu západ se rozsvítí zelená na semaforu sever, na semaforu západ a východ bude svítit červená (řádek 5.). Po zhasnutí zelené na semaforu sever se rozsvítí na tomto semaforu žlutá a na semaforech západ a východ bude svítit červená (řádek 6.). Dále se bude cyklus opakovat od řádku 1.

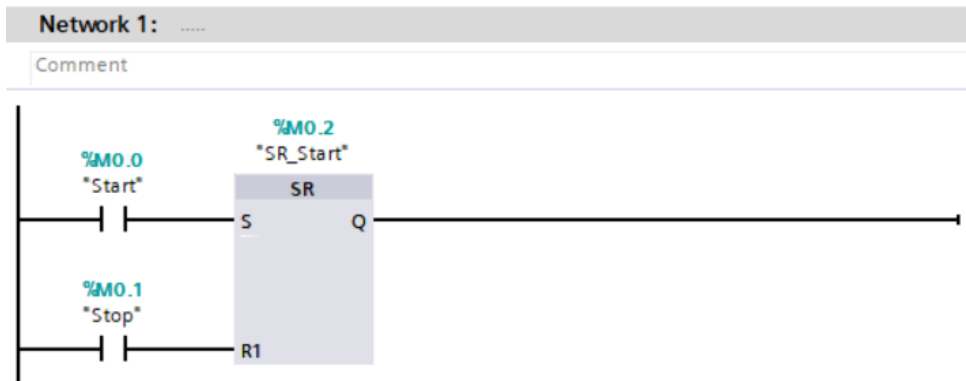
obPřehled adres používaných v programu: Adresy od č.19 „Východ červená“ mají adresu M z důvodu simulace na PC. Ve skutečném provedení by to byly adresy výstupů z S1200 „Q“. Adresy ba se musely v programu přepsat.

PLC tags				
	Name	Tag table	Data type	Address
1	Start	Default tag table	Bool	%M0.0
2	Stop	Default tag table	Bool	%M0.1
3	SR_Start	Default tag table	Bool	%M0.2
4	krok1_vychod_zelená	Default tag table	Bool	%M0.3
5	krok1_zapad_cervená	Default tag table	Bool	%M0.4
6	krok1_sever_cervené	Default tag table	Bool	%M0.5
7	krok1_východ_zluta	Default tag table	Bool	%M0.6
8	krok1_konec	Default tag table	Bool	%M0.7
9	krok2_západ_zelená	Default tag table	Bool	%M1.0
10	krok2_východ_cervená	Default tag table	Bool	%M1.1
11	krok_2_sever_cervená	Default tag table	Bool	%M1.2
12	krok2_západ_žlutá	Default tag table	Bool	%M1.3
13	Krok2_konec	Default tag table	Bool	%M1.4
14	krok_3_sever_zelená	Default tag table	Bool	%M1.5
15	krok_3_západ_cervená	Default tag table	Bool	%M1.6
16	krok_3_východ_cervená	Default tag table	Bool	%M1.7
17	krok_3_sever_žlutá	Default tag table	Bool	%M2.0
18	krok_3_konec	Default tag table	Bool	%M2.1
19	východ_cervená	Default tag table	Bool	%M2.2
20	východ_žlutá	Default tag table	Bool	%M2.3
21	východ_zelená	Default tag table	Bool	%M2.4
22	sever_cervená	Default tag table	Bool	%M2.5
23	sever_žlutá	Default tag table	Bool	%M2.6
24	sever_zelená	Default tag table	Bool	%M2.7
25	západ_cervená	Default tag table	Bool	%M3.0
26	západ_žlutá	Default tag table	Bool	%M3.1
27	západ_zelená	Default tag table	Bool	%M3.2

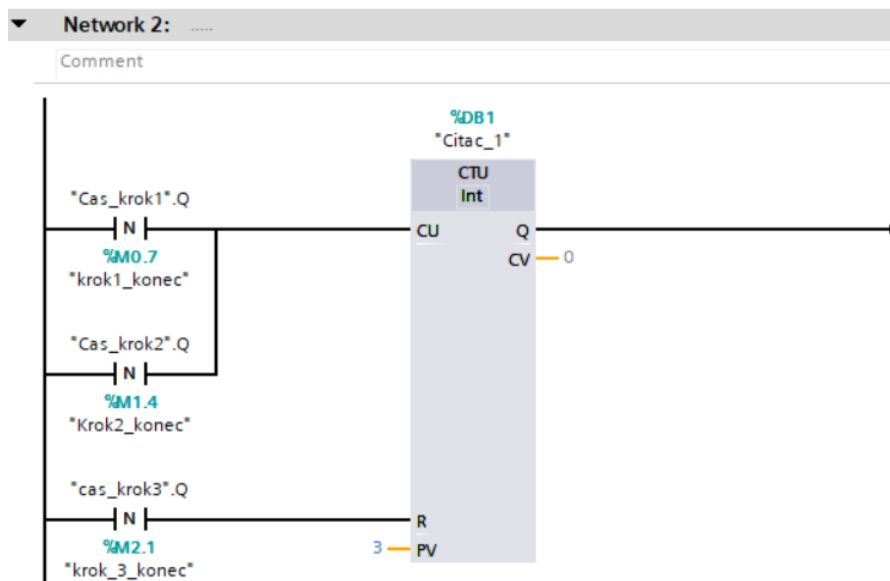
## Program

### Network 1:

Zde je vložen blok SR, který nahrazuje klasické zapojení stykače nebo relé ovládané dvěma tlačítky. Blok spínacího kontaktu „Start“ spíná výstup „Q“ bloku SR a blok spínacího kontaktu „Stop“ ho vypíná. Toto zapojení slouží k zapnutí cyklování semaforů nebo k jejich vypnutí.



### Network 2:



Blok „CTU“ je čítač pro zvyšování načítané hodnoty o 1. Čítač má za úkol posunout program z kroku 1. (obr.4) na krok 2. a při ukončení kroku 3. vynuluje čítač a postup se opakuje. V každém kroku je čítač nastavený na celkový čas kroku. Po ukončení časování se vypnutím časovače aktivuje kontakt N (impulz na sestupné hraně – při vypnutí). Čítač načítá od 0 do 3 (vstup PV nastavený na hodnotu 3). Po vypnutí výstupu časovače „Cas\_krok1.Q“ se hodnota čítače zvýší na 1, při vypnutí časovače „Cas\_krok2.Q“ na 2 a při vypnutí časovače „cas\_krok3.Q“ se čítač vynuluje (impulz na vstup R).

### Adresy výstupů M v jednotlivých krocích

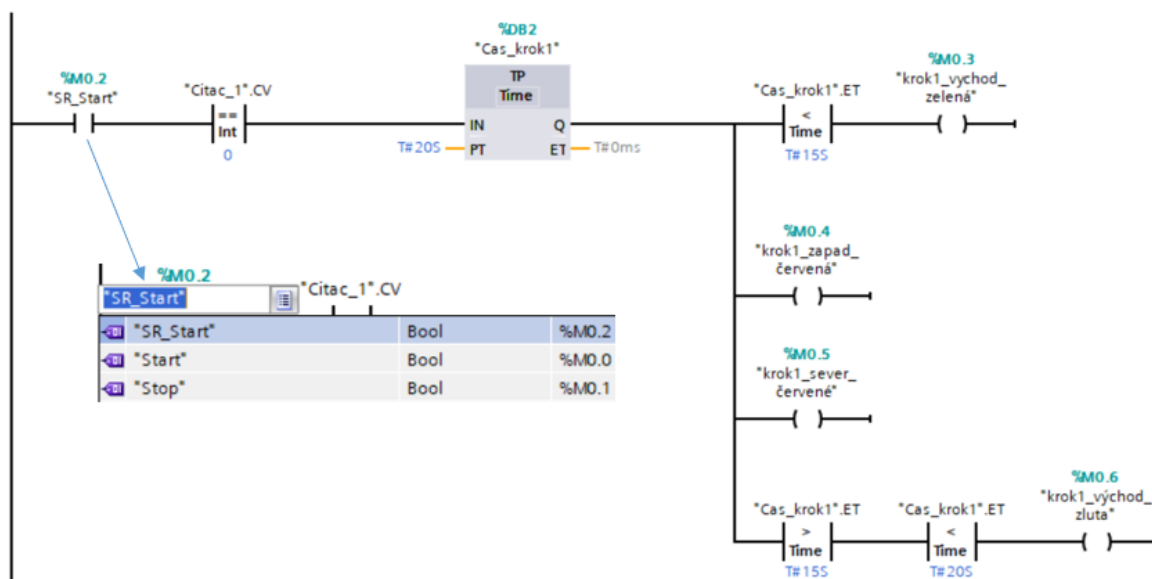
	čas	adresa
Krok 1	15s	krok1_východ_zelená
	20s	krok1_západ_červená
	20s	krok1_sever_červená
	5s	krok1_východ_žlutá

	čas	adresa
Krok 2	15s	krok2_západ_zelená
	20s	krok2_sever_červená
	20s	krok2_východ_červená
	5s	krok2_západ_žlutá

	čas	adresa
Krok 3	15s	krok3_sever_zelená
	20s	krok3_východ_červená
	20s	krok3_západ_červená
	5s	krok3_sever_žlutá

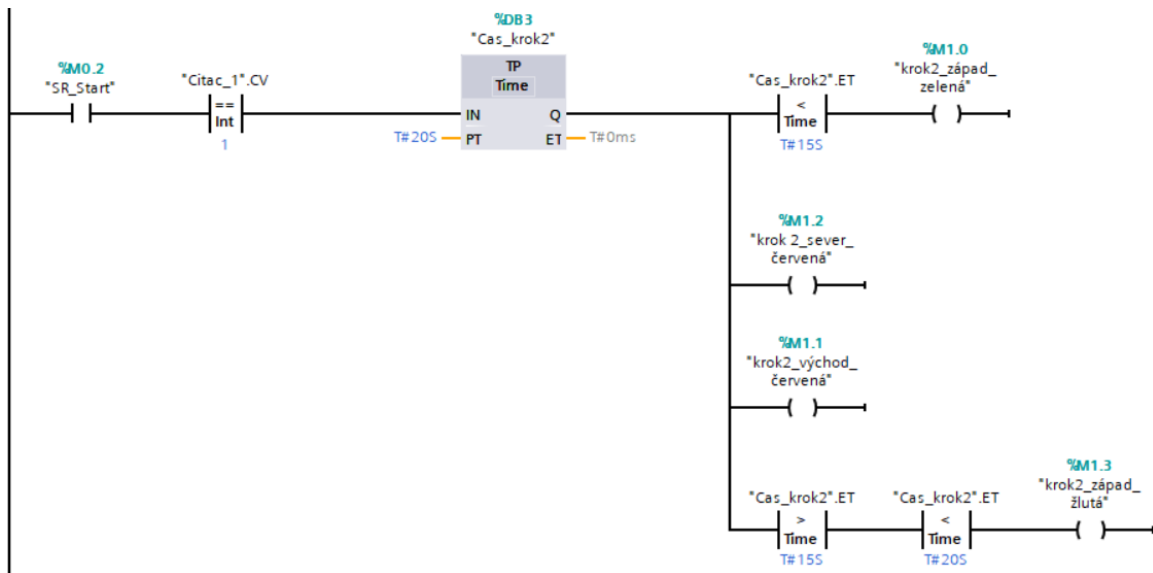
### Network 3:

spínací kontakt „SR\_Start“ je výstup bloku v Network 1. Po jeho sepnutí je hodnota čítače 0, ta se porovnává na komparátoru rovná se „=“ s nastavenou hodnotou, a to je 0. Protože se 0 = 0 spustí se časovač TP „Cas\_krok1“ nastavený na 20s. Na 20s se sepne červená na semaforu západ (výstup krok1\_zapad\_červená). Zelená „krok1\_vychod\_zelená“ začne svítit současně s červenými na jmenovaných semaforech, ale jen na 15s. To je dáno komparátorem „<“. Je-li čas menší než 15s zelená svítí, překročí-li 15s ta se vypne a zapne se žlutá, jejíž doba je omezena dvěma komparátory „>“ a „<“. Je-li doba spuštění časovače menší než 15s a současně větší než 20s je výstup „krok1\_východ\_zluta“ vypnutý. Je-li čas časovače větší než 15s a menší než 20s je výstup „krok1\_východ\_zluta“ sepnutý.



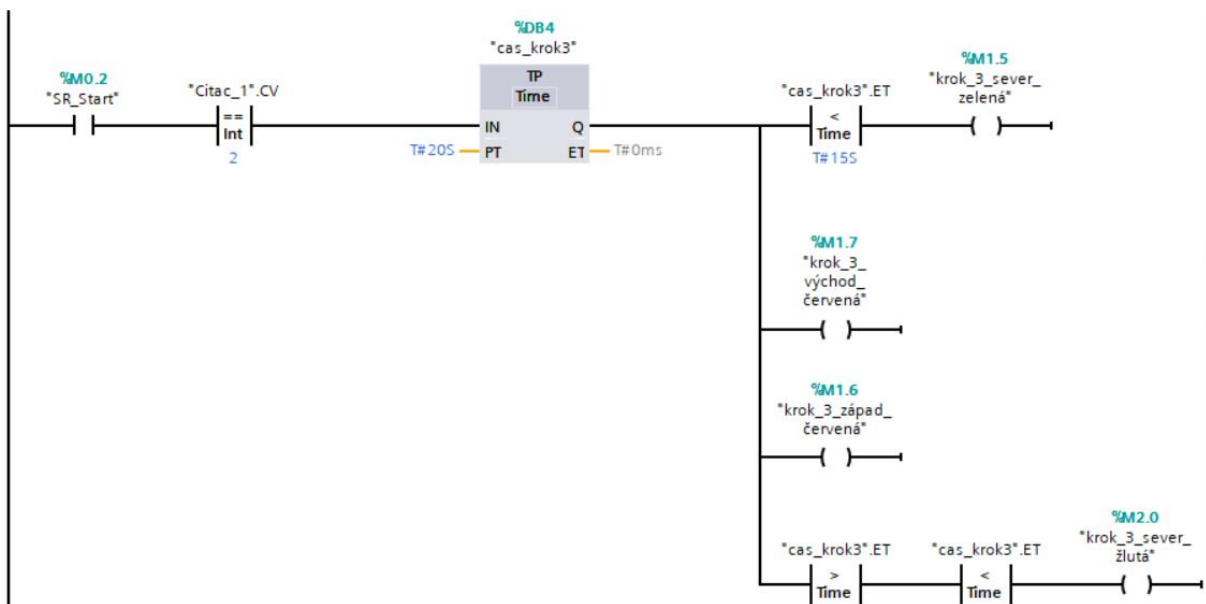
#### Network 4:

Zde je zapojení bloků shodné s rozdílem nastavení adres. Hodnota komparátoru „=“ je nastavena na 1 a porovnává s hodnotou na čítači, která se po vypnutí časovače „Cas\_krok1“ zvýšila o 1.



#### Network 5:

Zde je zapojení bloků shodné s předchozími Networky 3 a 4. Rozdíl je opět v nastavení adres. Hodnota komparátoru „=“ je nastavena na 2 a porovnává s hodnotou na čítači, která se po vypnutí časovače „Cas\_krok2“ zvýšila o 1, tedy na 2. Vypnutím výstupu časovače „cas\_krok3“ se resetuje čítač a protože je na něm 0, opakuje se cyklus od Network 3.



## Tabulka adres pro jednotlivé zapojení světel semaforů

### Semafor východ

svítí	adresa výstupu	adresa vstupu
červená	východ_červená	krok2_východ_červená krok3_východ_červená
žlutá	východ_žlutá	krok1_východ_žlutá
zelená	východ_zelená	krok1_východ_zelená

### Semafor západ

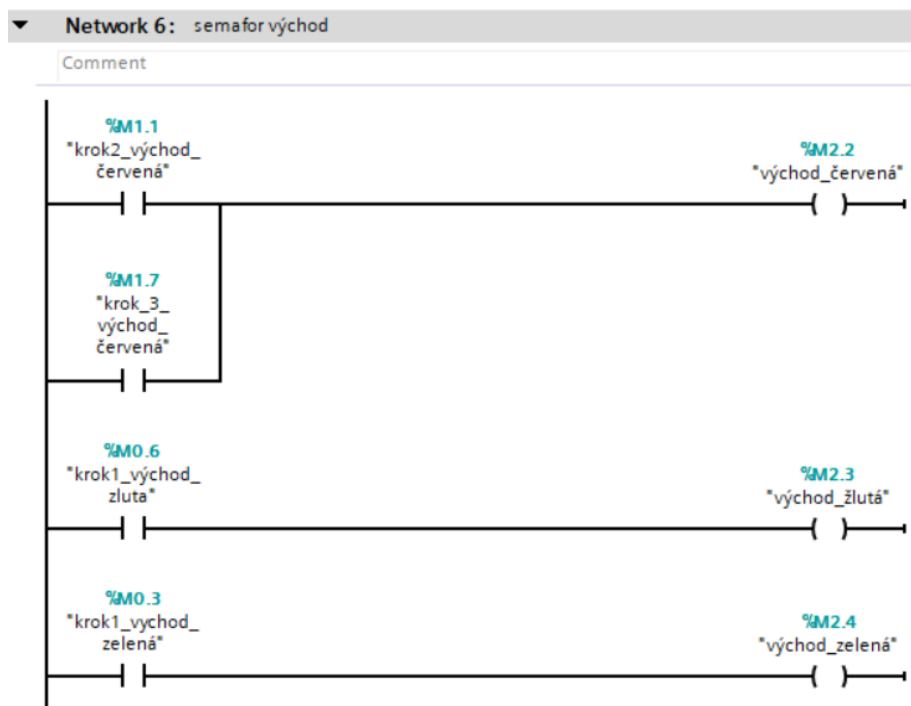
svítí	adresa výstupu	adresa vstupu
červená	západ_červená	krok3_západ_červená krok1_západ_červená
žlutá	západ_žlutá	krok2_západ_žlutá
zelená	západ_zelená	krok2_západ_zelená

### Semafor sever

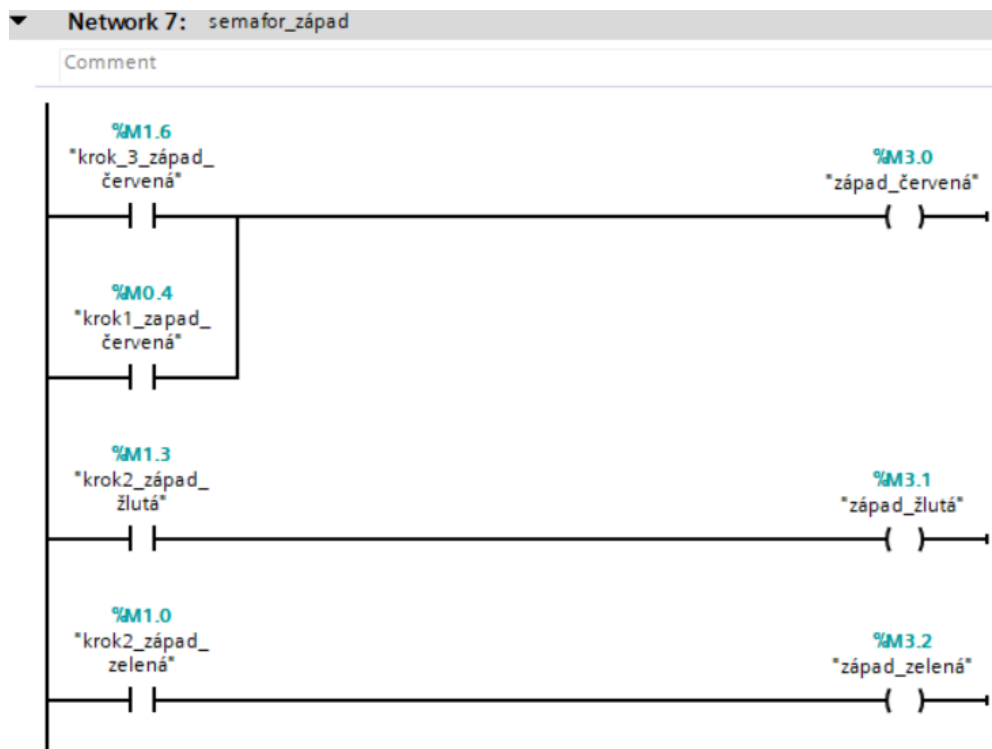
svítí	adresa výstupu	adresa vstupu
červená	sever_červená	krok1_sever_červená krok2_sever_červená
žlutá	sever_žlutá	krok3_sever_žlutá
zelená	sever_zelená	krok3_sever_zelená

Na následujících 3 Network se zapojí střídání barev na jednotlivých semaforem

### Zapojení semaforu východ



## Zapojení semaforu západ



## Zapojení semaforu sever

